

NASA Contractor Report 166079

Generalized Three-Dimensional Experimental Lightning Code (G3DXL) User's Manual

Karl S. Kunz

MAR 23 1989

*Publicly Released on
2/89*

Kunz Associates, Incorporated
Albuquerque, New Mexico 87191

Contract NAS1-16591

February 1986

(NASA-CR-166079) GENERALIZED
THREE-DIMENSIONAL EXPERIMENTAL LIGHTNING
CODE (G3DXL) USER'S MANUAL Contractor
Report, 1981 - 1986 (Kunz Associates)
145 p

N89-15779

Unclass

CSCI 04B G3/47 0192931



National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665

Date for general release February 28, 1989

CONTENTS

	<u>Page</u>
ABSTRACT	1
SECTION 1 - SCOPE AND OBJECTIVES	2
SECTION 2 - PROBLEM DEFINITION	3
2.1 THE PROBLEM	3
2.2 THE SOLUTION	3
2.3 NUMERICAL IMPLEMENTATION	4
2.4 PROGRAM ARCHITECTURAL DESCRIPTION	11
2.5 PROGRAM EXECUTION	14
2.6 LIMITATIONS	19
SECTION 3 - PROGRAMMING INFORMATION	21
3.1 PROGRAM FLOW	21
3.2 GLOBAL VARIABLES	28
3.3 DATA FILES	28
3.4 ERROR MESSAGES	28
3.5 SUBROUTINES	37
3.5.1 SUBROUTINE SETUP	37
3.5.2 SUBROUTINE BUILD	52
3.5.3 SUBROUTINE EADV	62
3.5.4 SUBROUTINE EBC	68
3.5.5 SUBROUTINE ERAD	82
3.5.6 SUBROUTINE HADV	91
3.5.7 FIELDS	96
3.5.8 SUBROUTINE SAVESB	101
3.5.9 SUBROUTINE OUTBND	106
3.5.10 SUBROUTINE DATASAV	114
3.5.11 SUBROUTINE PRINOUT	136

	<u>Page</u>
REFERENCES	140
APPENDIX - A	139
TABLES -	
1. Variables Input in the Main Program, Driver, of G3DXL	21
2. Global Variables	29
3. Variables Input in Subroutine SETUP	38
4. Local Variables - Subroutine SETUP	40
5. Local Variables - Subroutine ERAD	87
6. Variables Input in Subroutine DATASAV	114
7. Local Variables - Subroutine DATASAV	121

	<u>Page</u>
FIGURES - 1. Location of the Six Field Evaluation Points in a Typical Cell	7
1-a. Unexpanded F-106 Model	12
2. Expanded F-106 Model	13
2-a. G3DXL's Program Flow Using Four Pass Approach	22
2-b. First Pass, Diffusion Calculation	23
2-c. Second Pass, Expanded Diffusion Calculation	24
3. Flow Chart for Subroutine SETUP	39
4. Flow Chart for Subroutine BUILD	43
5. Flow Chart for Subroutine EADV	64
6. Determining E-Field Boundary Condition	69
7. Generalized Flow Chart of Subroutine EBC	71
8. Flow Chart for Subroutine EBC, Perfectly Conducting Portion Only	72
9. Field Components at the Outer Boundary of the Cell Space	83
10. Flow Chart for Subroutine ERAD	88
11. Flow Chart for Subroutine HADV	93
12. SAVESB Flow Chart	102
13. OUTBND Flow Chart	107
14. Flow Chart for Subroutine DATASAV	115
15. Distribution of Field Component Locations About Test Points	116
16. Example Field Component Arrangement about a Test Point	118
17. Interpolated Field Values about a Test Point	119
18. Interpolated Field Values Used to Extrapolate to Test Point Location	119
19. Graphical Description of THETA	120
(Angle defining sensor rotation).	

ABSTRACT

The computer code G3DXL computes time domain scattered fields and surface currents and charges induced by a driving function on and within a complex scattering object which may be perfectly conducting or a lossy dielectric. This is accomplished by modeling the object with cells within a three-dimensional, rectangular problem space, enforcing the appropriate boundary conditions and differencing Maxwell's equations in time. In the present version of the program, the driving function can be either the field radiated by a lightning strike or a direct lightning strike. The scattering object is the F-106 B aircraft.

The program has been successfully executed on the CDC 203 at NASA Langley where it is currently operational.

SECTION 1

SCOPE AND OBJECTIVES

This document presents information concerning the programming, maintenance and operation of the computer code G3DXL. This information is intended to facilitate 1) effective use of the program as is and 2) conversion and modification of the code for other test items and excitation sources.

The theoretical basis for the code is discussed, the numerical implementation of the theory is presented, and the limitations are enumerated. This discussion is followed by a presentation of the program architecture, including flow charts, global variable definitions, input and output requirements. A discussion of each individual subroutine is given which includes a subroutine description, local variable definition (where needed), flow chart, and a listing of the subroutine. The reader is provided in this manual with an understanding of the theoretical basis of the code and its practical implementation.

SECTION 2

PROBLEM DEFINITION

2.1 THE PROBLEM

G3DXL addresses the problem of calculating in the time domain the currents and charges induced on and within a composite aircraft, along with the resulting scattering fields, by an excitation source, in this case a direct lightning strike or the radiated lightning field. The scatterer may be imbedded in a homogeneous media (an aircraft in flight, for example) or may be located over a lossy ground plane (an aircraft over a runway, for example).

2.2 THE SOLUTION

G3DXL finds the induced currents and charges on the scatterer along with the scattered fields by differencing Maxwell's equations in time using an algorithm first described by Yee [1]. This algorithm is applied to a problem space with a mesh of dimensions of $29 \times 29 \times 29$. Even larger problem spaces on the order of $50 \times 50 \times 50$ are possible on certain large computers, such as the DEC VAX 11/780. The problem of echoes from the finite problem space's outer boundaries is circumvented by the application of Merewether's radiating boundary condition [2], which is based on the fact that, far from the scatterer, the scattered fields must behave as $f(\theta, \phi)g(t-r/c)/r$. Here, $f(\theta, \phi)$ represents an arbitrary variation in θ and ϕ , $g(t-r/c)$ expresses the retarded time behavior of the fields and the $1/r$ describes the far field dependence on r . It is the $1/r$ dependence that is of important here, as it allows fields near the outer boundary to be extrapolated to the outer boundary, thereby affecting the outer radiating boundary condition.

2.3 Numerical Implementation

The three-dimensional, finite-difference formalism employed by G3DXL separates the fields into incident and scattered fields and solves directly for the scattered fields, using Maxwell's equations. The incident field is used only to establish the value of the scattered field throughout the scatterer's volume, where the fields are related by what will be referred to here as a volume boundary condition.

The formalism (Appendix B gives more details) starts with Maxwell's equations for the total fields:

$$\nabla \times \vec{E} = -\mu \frac{\partial \vec{H}}{\partial t}$$

$$\nabla \times \vec{H} = \epsilon \frac{\partial \vec{E}}{\partial t} + \sigma \vec{E}.$$

Subtracting the Maxwell equations for the incident field propagating in free space, i.e. the free space form of the Maxwell equations:

$$\nabla \times \vec{E}^i = -\mu_0 \frac{\partial \vec{H}^i}{\partial t}$$

$$\nabla \times \vec{H}^i = \epsilon_0 \frac{\partial \vec{E}^i}{\partial t}$$

yields the generalized equations:

$$\mu \frac{\partial \vec{H}^s}{\partial t} = -\nabla \times \vec{E}^s - (\mu - \mu_0) \frac{\partial \vec{H}^i}{\partial t}$$

$$\epsilon \frac{\partial \vec{E}^s}{\partial t} + \sigma \vec{E}^s = \nabla \times \vec{H}^s - \sigma \vec{E}^i - (\epsilon - \epsilon_0) \frac{\partial \vec{E}^i}{\partial t}$$

The generalized equations, as they stand, are completely general and may be applied over all space. In free space, the terms with $(\mu-\mu_0)$, $(\epsilon-\epsilon_0)$ and σ as multipliers vanish yielding the free space form of the equations. On a perfectly conducting surface, where $\sigma=\infty$ the generalized equations reduce to:

$$E_T^S = -E_T^I$$

where T refers to the tangential components of the E-field on the perfectly conducting surface.

Prior work with finite difference codes typically used the free field formulation with the perfectly conducting boundary condition. G3DXL uses the generalized equations throughout the volume of a lossy dielectric scatterer, the free field equations in free space and the perfectly conducting boundary condition for a perfect conductor. This approach was deemed more efficient than the use of the generalized equations throughout all space.

The actual conversion of the differential equations into a finite difference form will be presented here for a perfectly conducting scatterer about which the free space Maxwell equations for the scattered field hold, i.e.,

$$\nabla \times \vec{E}^S = -\mu \frac{\partial \vec{H}^S}{\partial t}$$

$$\nabla \times \vec{H}^S = \epsilon \frac{\partial \vec{E}^S}{\partial t} + \sigma \vec{E}^S$$

but are written in a finite-difference form using two point central differencing in a rectangular coordinate system. For example:

$$\frac{\partial H_z^{\text{scat}}}{\partial y} - \frac{\partial H_y^{\text{scat}}}{\partial z} = \frac{\partial E_x^{\text{scat}}}{\partial t} + \sigma E_x^{\text{scat}}$$

becomes

$$\begin{aligned}
& \frac{(H_z^{\text{scat}})^n(I, J, K) - (H_z^{\text{scat}})^n(I, J-1, K)}{Y(J) - Y(J-1)} \\
& - \frac{(H_y^{\text{scat}})^n(I, J, K) - (H_y^{\text{scat}})^n(I, J, K-1)}{Z(K) - Z(K-1)} \\
& = \epsilon \frac{(E_x^{\text{scat}})^{n+1/2}(I, J, K) - (E_x^{\text{scat}})^{n-1/2}(I, J, K)}{\Delta t} \\
& + \sigma \frac{(E_x^{\text{scat}})^{n+1/2}(I, J, K) + (E_x^{\text{scat}})^{n-1/2}(I, J, K)}{2}
\end{aligned}$$

where Δt , $Y(J) - Y(J-1)$, etc. are discrete time and spatial increments used in the calculation. The superscript n is the time index and I, J, K label the nodes in the three-dimensional mesh of the problem space. Similar expressions can be derived for the remaining five equations, involving the remaining components. Note that the H-field components at times $(n+1)\Delta t$ can be evaluated from their own value at the earlier time $n\Delta t$ and from the E-field components at the earlier times $(n+1/2)\Delta t$. Similarly the E-field components at times $(n+1/2)\Delta t$ are evaluated from the H-field components at times $n\Delta t$ and from their own values at earlier times $(n-1/2)\Delta t$. This results in a fully explicit and automatically central differencing scheme that reduces the truncation or round-off error [3,4].

The difference equations are applied, at successive discrete time steps, to a cell space; in this case, a three-dimensional, rectangular cell space. That is, the equations are assumed to hold for all the cells of the space. A single cell and the field components associated with it are shown in Figure 1. The size of the individual cell determines the time step Δt . For stability, Δt must satisfy the Courant stability condition, which may be expressed as [5].

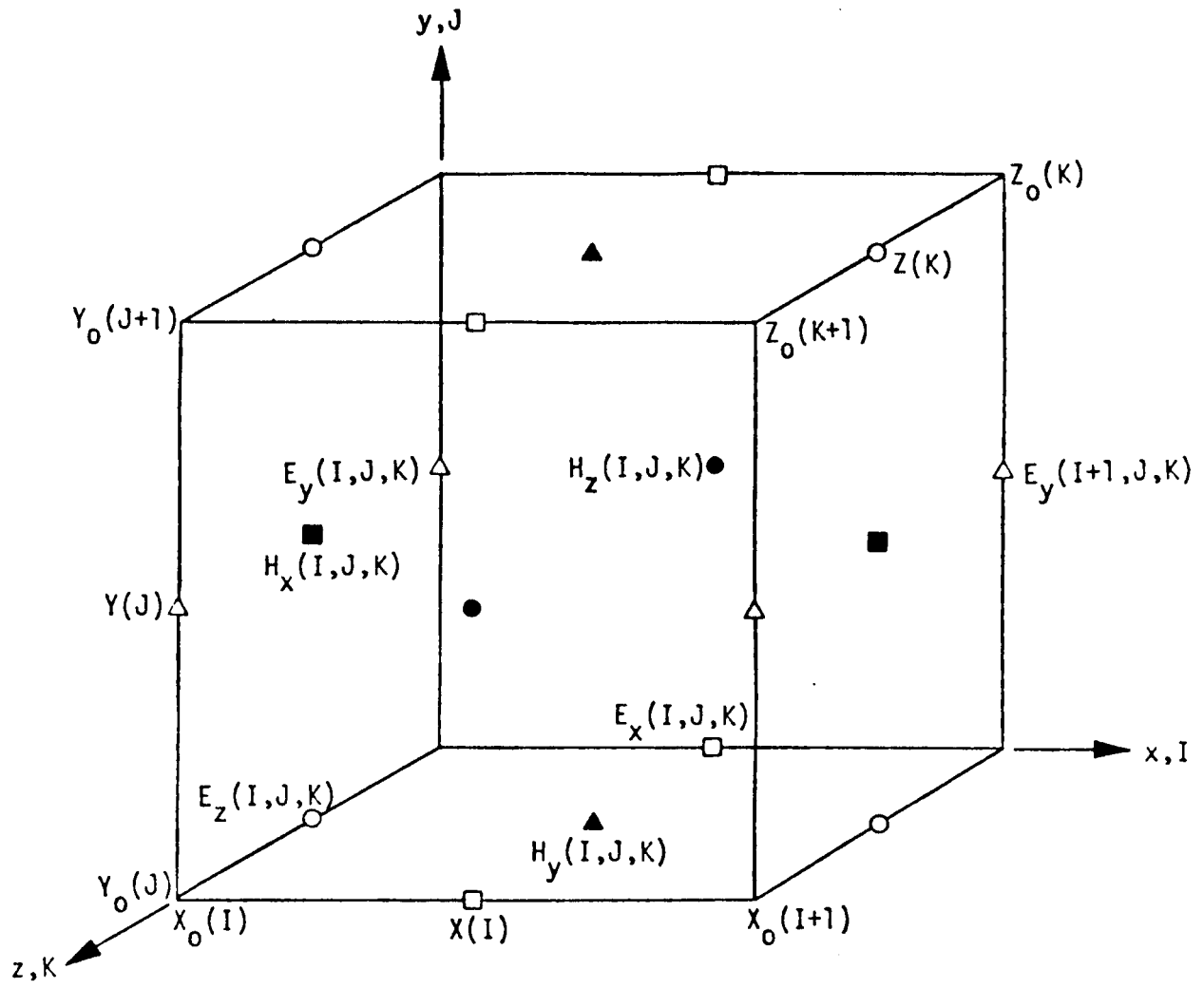


Figure 1. Location of the Six Field Evaluation Points in a Typical Cell

$$\Delta t = \min_{I,J,K} ((X_O(I+1) - X_O(I))^{-2}(\epsilon\mu)^{-1} \\ + (Y_O(J+1) - Y_O(J))^{-2}(\epsilon\mu)^{-1} + (Z_O(K+1) - Z_O(K))^{-2}(\epsilon\mu)^{-1})^{-1/2}$$

If ϵ or μ are inhomogeneous, their dependence on (I,J,K) must be included in evaluating the minimal time steps of the above equation. If the time step is greater than Δt , the predicted responses grow without bound. Note that Δt is approximately the transit time across the cell. The cell size also determines the frequency range of valid data. When a quarter wavelength of the frequency component of interest is smaller than the cell dimensions, the answer returned is in error.

Two boundary conditions must be applied to this cell space. The first defines the object for which the scattered fields are to be calculated. We shall only consider a perfect conductor here for simplicity. For each cell face that defines a portion of aircraft surface, the total tangential E-field vanishes so that

$$E_{\text{tan}}^{\text{scat}}(t) = -E_{\text{tan}}^{\text{inc}}(t)$$

At $t=0$, $E^{\text{inc}}(t)$ is assumed zero and so $E^{\text{scat}}(t)$ is also zero. At later times the source or driving field $E_{\text{tan}}^{\text{inc}}(t)$ appears at the scatterer's surface with non-zero value determining $E_{\text{tan}}^{\text{scat}}(t)$ and, through the difference equations, determining the scattered fields throughout the cell space.

The second boundary condition is on the outer surface of the cell space where the radiation condition is imposed [2]. This boundary condition requires the scattered E-field to behave as $E^{\text{scat}}(t) = f(\theta, \phi) g(t-r/c)/r$ at the outer surface, the same field behavior as occurs far from the scatterer. By imposing this behavior on the surface of the cell space, a finite cell space can be made to approximate an infinite cell space. In effect, the outermost E-

fields are made to behave nearly the same as they would have had the cell space continued on beyond the outermost cells indefinitely, leaving the inner cells seemingly imbedded in an infinite cell space. As a rule of thumb, this approximation is successful when the cell space is at least twice the scatter's dimensions in each direction.

As an example of this procedure, let J_m be the maximum index in the y-coordinate for the cell space. The radiation condition applied to E_z^{n+1} in the y-direction is:

$$E_z^{n+1}(I, J_m, K) = \frac{R_z(I, J_m - 1, K)}{2R_z(I, J_m, K)} * [\theta_{zy}(\theta_{zy} - 1)E_z^{n-1}(I, J_m - 1, K) \\ + 2(1 - \theta_{zy}^2)E_z^n(I, J_m - 1, K) + \theta_{zy}(\theta_{zy} + 1)E_z^{n+1}(I, J_m - 1, K)]$$

where

$$R_z(I, J_m, K) = \sqrt{X_o(I)^2 + Y_o(J_m)^2 + Z_o(K)^2}$$

$$\theta_{zy} = 1 - \frac{R_z(I, J_m, K) - R_z(I, J_m - 1, K)}{c\Delta t}$$

and $X_o(I)$, $Y_o(J)$, $Z_o(K)$ are coordinate values for the points considered, c is the velocity of light. As can be seen, the fields at the outer boundary are found from interpolating, in time, the fields that are one cell in from the outer boundary. The times are the present program cycle, and the two prior program cycles. This interpolation is based on a constant cell size. If an expanding mesh is used, a somewhat more complex expression must be employed [5].

Similar conditions (there are eleven more) can be written down for E_x , E_y , and E_z at the six surfaces of the box.

By assigning appropriate constitutive parameters (ϵ , μ , and σ) to the first few tiers of cells at the base of the cell space, a ground can be modeled from which the scattered fields are partially reflected. Since the radiation condition is applied to all six faces, only the desired ground reflection will be present; the fields transmitted into the ground will not be reflected back from the outer surface of the cell space.

The cell size used need not be uniform. By enlarging the cell size outside the immediate region of the aircraft, the outer boundary can be further removed from the aircraft. (Expansion factors much greater than 1.3 are not recommended. The reason is that severe discontinuity effects can be seen.) Reflections from the outer boundary, that occur because the radiation condition is imposed at moderate distance R from the origin, are thereby lessened.

The surface current density and surface charge density at any position on the aircraft are found from

$$\bar{J} = \bar{n} \times (\bar{H}^{\text{scat}} + \bar{H}^{\text{inc}}) \text{ and } \rho_s = E_{\text{normal}}^{\text{scat}} + E_{\text{normal}}^{\text{inc}}$$

respectively, where \bar{n} is the outward normal of the scatterer's surface. The values of the scattered fields at the position of interest are found from the adjacent nodal point values using a combination of interpolation and extrapolation. These time-domain responses can be transformed to yield frequency domain data over a broad frequency range.

2.4 Program Architectural Description

G3DXL is designed to predict exterior and interior responses on complex scattering objects, such as aircraft. To achieve the needed degree of spatial resolution for interior coupling predictions the expansion technique is employed. A preliminary run with a coarsely modeled aircraft (Figure 1-a) is used to obtain the response about a portion of the aircraft. This response imposed on the outer boundary of the problem space allows a second run to be made in which only the portion of the aircraft is modeled (Figure 2), but in finer detail and with the entire aircraft response preserved. The use of the expansion technique requires what is, in effect, two separate runs.

The scattering object can also be a perfect conductor, a restriction that some prior finite difference codes, such as THREDE [5,6], had to adhere to; or it could be a lossy dielectric, as in a composite aircraft. This last and more general case is treated using the generalized finite difference technique as embodied in G3DXL. A thin plate algorithm (Appendix A) even allows composite panels to be treated. The generalized finite difference technique and thin plate algorithm do not require more runs, but each run using them is more complex. Such a run must be able to treat perfectly conducting solids and surfaces or plates and must be able to treat lossy dielectric solids and plates.

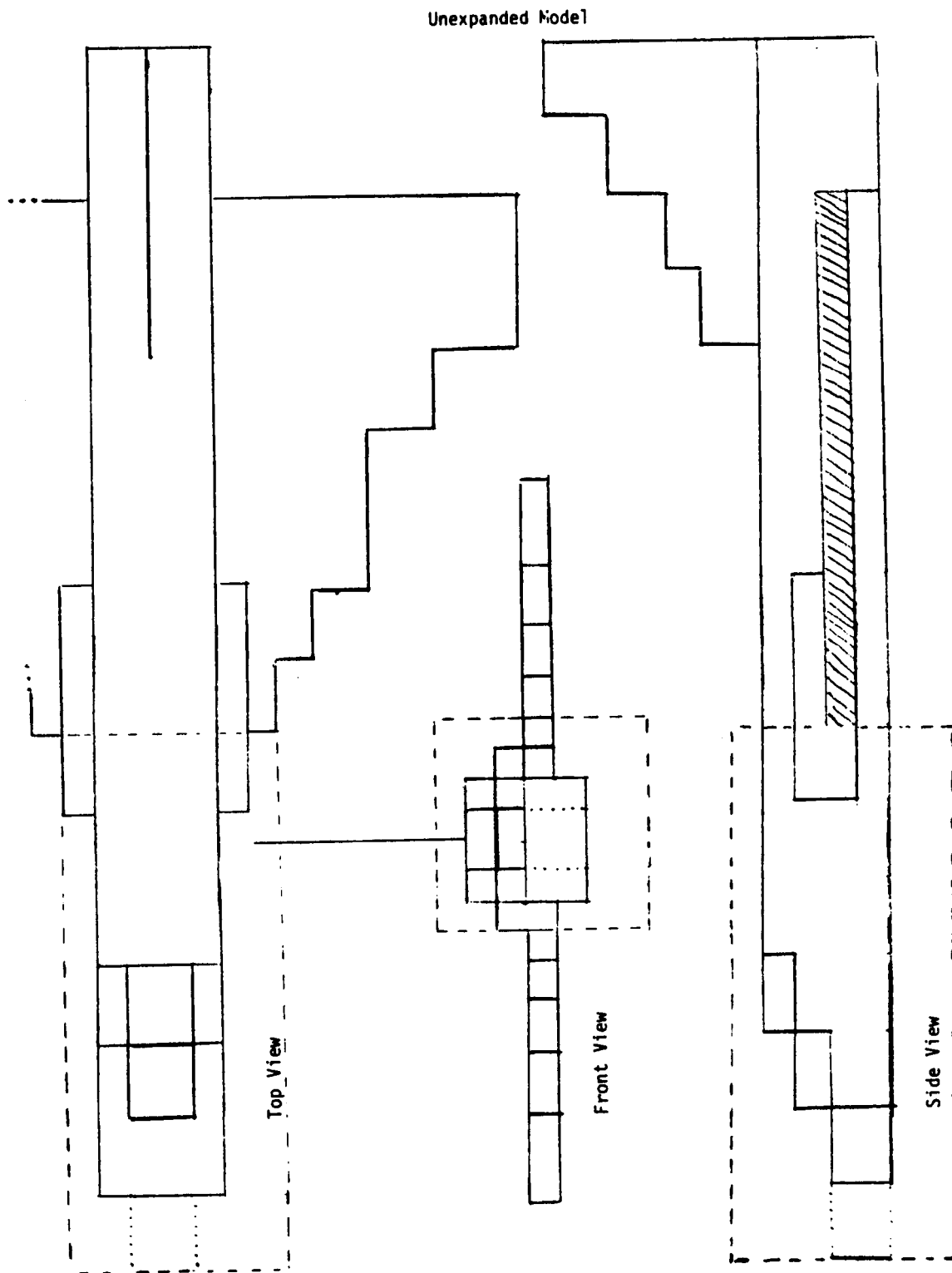


Figure 1-a. Unexpanded F-106B Model

ORIGINAL PAGE IS
OF POOR QUALITY

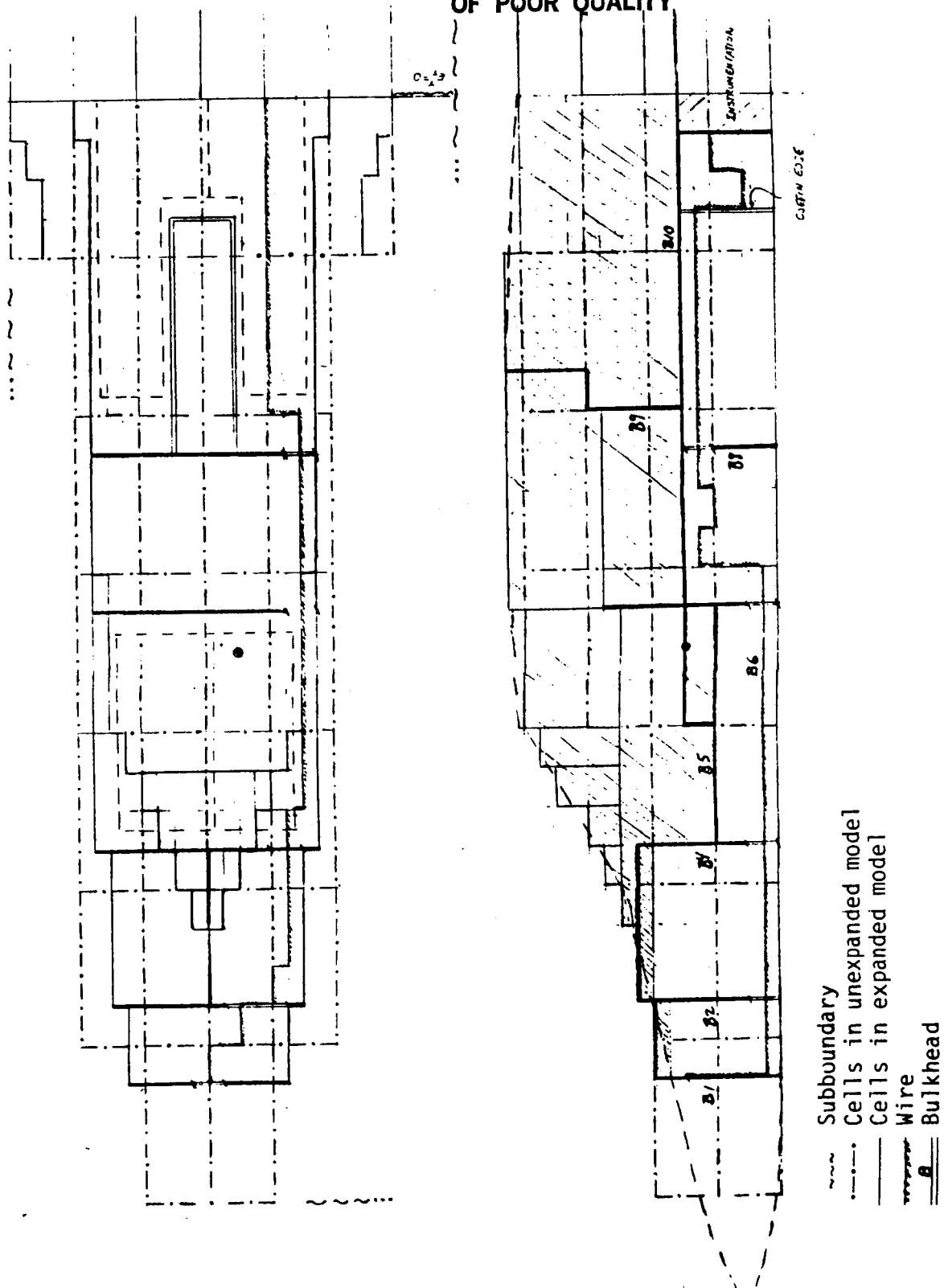


Figure 2. Expanded F-106 Model

2.5 Program Execution

The following steps are a concise guide to the execution of the code.

- 1) Always maintain two original backup copies: the first, a direct access file resident in the computer's memory and the second, a card deck, the same as the file, resident with the user in charge of maintaining the code.
- 2) From the original file in the computer, make a new file under a new name. It is this file that will be modified (using the XEDIT capability) to model the correct problem.
- 3) When the new file is modified and running properly, it is prudent to make a copy of the file. This copy serves as a secondary backup. If further modifications are made and they result in a nonrunning program, one can always return to the secondary backup and try again.
- 4) Modifications to the original code, as embodied in the original backup, can be made only after the selection of an excitation source and an interaction object.
- 5) The excitation source can be either a radiated lightning field or a direct lightning strike. To select one or the other, the parameter IDLS is set equal to 0 for a radiated lightning field or 1 for a direct lightning strike. IDLS appears as a data statement in subroutine SETUP.
- 6) The excitation source specifications are set as follows. For a radiated lightning field a double exponential waveform $A(e^{-\alpha t} - e^{-\beta t})$ is assumed. The parameters A, α and β are represented by AMP, ALPHA and BETA in the code. They are set in subroutine SETUP. Current values are AMP = 5.92E4, ALPHA = 4.08E6, BETA = 1.0E8.

7) The actual calculation of the radiated lightning fields is performed in the functions EINCX(I,J,K),...,HINCZP(XP,YP,ZP). At present the fields are for broadside incidence from above. The electric field points in the x direction and the magnetic field points in the z direction.

8) The interaction object, typically an aircraft, must be modeled within the problem space boundaries set by a space of 28 x 28 x 28 cells. Further, the outer radiating boundary condition can be expected to work reasonably well only if the interaction object occupies, at most, 20 cells across the problem space in any direction. Thus, the interaction object can be divided by no more than 20 in any linear dimension. If one dimension is L, then the cell used in the problem space would have dimension L/20 in this direction. A typical aircraft is on the order to 20 meters long. Thus, cells approximately 1 meter long are used in the M=1 loop where the entire aircraft must be modeled. Generally, cells on the order of 1m x 0.5m x 0.5m are used. This allows for a reasonably accurate rendition of the aircraft fuselage. Expanding the cells where the wings lie is, however, necessary. This allows all of the wings to be modeled in the problem space at a sacrifice in detail in the region of the wings, where it is often not needed.

9) For the M=1 loop, when the entire interaction object must be modeled, a set of scale drawings or the equivalent are needed. They define the interaction object geometry. Additionally, the constitutive parameters (σ and ϵ) of the material comprising the interaction object are needed along with the material thickness. The model is input into the code by:

- o setting the cell size ($\Delta x, \Delta y, \Delta z$) through the parameters DELX, DELY and DELZ in subroutine SETUP.
- o setting the expansion rates of the cells (XPANX, XPANY, XPANZ) and the indices where cell expansion starts, (IUP,...KDOWN) in subroutine SETUP.
- o setting the NOPE(I,J,K) array in subroutine BUILD. Nonzero values of NOPE(I,J,K) place a material media in the (I,J,K)th cells. When NOPE(I,J,K) = 0 the cell has the properties of free space. The option exists to select a perfectly conducting material that fills

the cell ($\text{NOPE}(I,J,K) = 4$) or forms a plate on one side of the cell ($\text{NOPE}(I,J,K) = 1$ for the (Y-Z) plane, 2 for the (X-Z) plane and 3 for the (X-Y) plane). (All sides are the near side of the cell as defined along the x,y,z axis that follow the increase in the indices I, J, and K respectively.) The option also exists to replace the perfectly conducting material with a lossy dielectric material. The same approach is used as for the perfectly conducting material, except 1,2,3,4 are replaced by 5,6,7,8 in the values that may be assigned to $\text{NOPE}(I,J,K)$. Cells are set to nonzero values so as to fill the problem with material following the geometry of the interaction object and having the same material properties (ϵ and σ).

- o setting ϵ and σ for any lossy dielectric material in the interaction model via a data statement for EPS and SIGMA in the subroutine SETUP.
- o setting plate thickness TX for lossy dielectric plates in the (Y-Z) plane, TY for lossy dielectric plates in (X-Z) plane and TZ for lossy dielectric plates in the (X-Y) plane. (Presently EPS, SIGMA, TX, TY and TZ do not vary as a function of position). At a large cost in memory, but within the capabilities of the Cyber 203, these parameters can be input as a function of position, i.e. $\text{EPS} \rightarrow \text{EPS}(I,J,K)$, etc. If this change is required, then an array, such as $\text{EPS}(I,J,K)$, must be specified in subroutine SETUP and the CDC XEDIT capability used to replace EPS by $\text{EPS}(I,J,K)$ everywhere, including common statements. Related quantities to EPS, such as EPEFFX , EPEFFY , EPEFFZ , EXPON , EXPDX , EXPDY , EXPDZ , XEXP , XEXPX , XEXPZ , and to SIGMA, such as RSIGMA , must also be written as three dimensional arrays and dimensioned as such or, more appropriately, written out explicitly in terms of $\text{EPX}(I,J,K)$, $\text{SIGMA}(I,J,K)$, $\text{TX}(I,J,K)$, etc. at the cost of an increased computational burden with the savings of a very large amount of memory.

10) Interaction object models need only be constructed for those loops desired, i.e.

M=1 for unexpanded model where only surface responses are required;

M=2 for expanded model where only aperture coupling and diffusion may be treated;

11) The desired permutation of the two loops is set by MM. Allowed values are MM=1, 12, where, for example, MM=12 results in the M=1 and 2 loops being exercised. MM is set in program DRIVER.

12) The number of time steps, N , must also be set in DRIVER. Dimension restrictions limit N at present to 2000. Further, the expansion factor EXPFAC times N for the $M=1$ loop should yield N for the $M=2$ loop. Thus, when an expanded run is made, $N \leq 2000$ and $N \leq 2000/\text{EXPFAC}$ ($2000/4 = 500$ at present) for unexpanded runs made in conjunction with an expanded run.

13) For expanded runs ($M=2$) the necessary subboundary information on the preceding unexpanded run ($M=1$) is saved via subroutine SAVESB and interpolated for use in the expanded run in subroutine OUTBND. Presently, the expansion factor, EXPFAC, is four, so that the space surrounded by the subboundary on the unexpanded run is $7 \times 7 \times 7$ cells. The use of interpolation only and the positioning of the tangential E field components on the subboundary faces requires that an array of 8×9 field components be saved on each face. As there are two components on each face and six faces, a total of 864 components are saved in each time step in the array ARAY, which has dimensions of (864,500). For different expansion factors the 864 would have to change in ARAY, which appears in subroutines SAVESB and OUTBND. Also, CRAY(8,9) appearing in subroutine OUTBND for interpolation purposes would have to be changed, as would the auxiliary array ARRAY.

14) The location of the subboundary is set by a data statement in INEAR,...,KFAR in subroutines SAVESB and OUTBND.

15) Interior regions are typically modeled for the $M=2$ loop where sufficient resolution is available. Wires are electrically conducting ($E^{\text{scat}} = -E^{\text{inc}}$) for $M=2$. They are manually specified in EBC.

16) Lightning channels are modeled as exterior wires running from the outer surface of the problem space to the scattering object. They are electrically conducting for $M=1$ and 2 and are manually specified in EBC. A loop of H -field components drive the electrically conducting lightning channel for $M=1$ and is specified manually in HADV. For $M=2$ the loop is not presented, only the channel. This is because the loop acts as a source and must be outside the subboundary volume for the expansion technique to work.

17) Test point locations are selected manually in DATASAV. The desired locations (up to 24) are specified by XOBS(NPT), YOBS(NPT), ZOBS(NPT), which are the test point locations in meters from the center of the problem space. Setting NPLANE(NPT) for each location determines the surface on which test point is located or, more precisely, the direction from which the fields are extrapolated to the test point. NPLANE=1 selects a test point on a Y-Z plane with the fields extrapolated in from left to right; 2 is the same, except the extrapolation is from right to left. NPLANE=3 selects a test point on a X-Z plane with the fields extrapolated from high to low; 4 is the same except the extrapolation is from low to high. NPLANE=5 selects a test point on a X-Y plane with the fields extrapolated from back to fore; 6 is the same except the extrapolation is from fore to back. A test location on a surface can be for an exterior or interior response depending on the selection of NPLANE, which determines from what direction the extrapolation is made. THETA(NPT) aligns the test point response components, that is for THETA=0, HSTOR1 is the axial current for the test point on the sides of a cylinder and HSTOR2 is the circumferential current. When THETA=90, this assignment is reversed. For THETA somewhere in between, HSTOR1, for example, is a mix of axial and circumferential currents. For most applications THETA can be left at 0.

18) To obtain current responses for late times economically, the data records for the HSTOR1 component are extended analytically LMAX * IP time steps. IP is the number of time steps taken before response data is saved. For example, for IP=2 data is saved every other time step. Both LMAX and IP are set via a data statement in subroutine SETUP. For currents on interior wires a loop of magnetic field about the wire can be manually specified and output as CSTORE. NPLANE must be selected so that each component is stored in HSTOR1, which is analytically extended. The, CSTORE is also of extended duration and the late time currents are known.

There are three limitations on the operation of G3DXL which are imposed by the mathematical model and the computer facilities. These are listed and discussed below.

- 1) Run time is limited to ≈ 2000 program cycles for a single loop for typical aircraft scattering problems. A program cycle is typically on the order of 1 ns resulting in 500 to 2000 ns of data. Instabilities in the radiation boundary condition return a growing oscillation in the late-time data that overwhelms the true response. This limitation is lessened by maintaining nearly equal cell sizes on the outer boundaries (a condition that is automatically met when using a constant mesh, but not always when using an expanding mesh.) Differences of greater than a factor of two should be avoided.
- 2) The model is limited in size to be no more than one-half the cell space in any direction. If the model is not restricted to this size limit, the approximation upon which the radiating boundary condition is based, namely that the scattered fields behave as far fields at the outer boundary, is severely compromised and reflections off the outer boundary will be excessively large.
- 3) The upper frequency is limited to $c/4\ell$, where ℓ is the largest dimension of the cells used to define the scatterer. This is equivalent to quarter wavelengths equal to the largest cell dimension. At these wavelengths, the granularity in the scattering model becomes apparent and at shorter wavelengths, i.e., high frequencies, invalidates the data.

- 4) Only expansion factors of 2, 4, 7, and 14 are allowed, as these are the only integral factors possible with a 28-on-a-side cell space. Further, the factor of 14 is not recommended, as it will involve very crude subboundary field interpolations.

SECTION 3

PROGRAMMING INFORMATION

This section provides information directed toward implementation of the computer program G3DXL on a particular computer, modification or extension to meet particular needs or conversion for a different computer environment.

3.1 PROBLEM FLOW

Flow charts showing the overall program structure for G3DXL are given in Figures 2a-c. The functions performed by each subroutine shown on the chart are described in detail in Section 3.5. The main program inputs a variable indicating how frequently to output results; it sets the starting and ending time for the computations and also increments time. Most importantly, in addition, it calls subprograms SETUP, BUILD, EADV, HADV, SAVESB, and DATASAV. The variables input in the main program are described in Table 1.

Table 1. Variables Input in the Main Program, DRIVER, of G3DXL

Variable	
<u>Name</u>	<u>Description</u>
MM	Desired permutation of the four possible configurations
IDLS	0=Indirect Radiation 1=Direct Lightning Strike
TSTART	Time (in seconds) at which to start computations

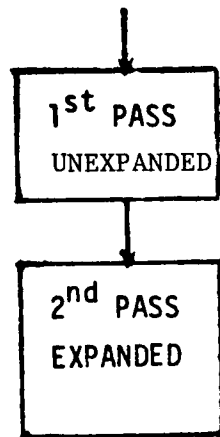


Figure 2a. G3DXL's Program Flow

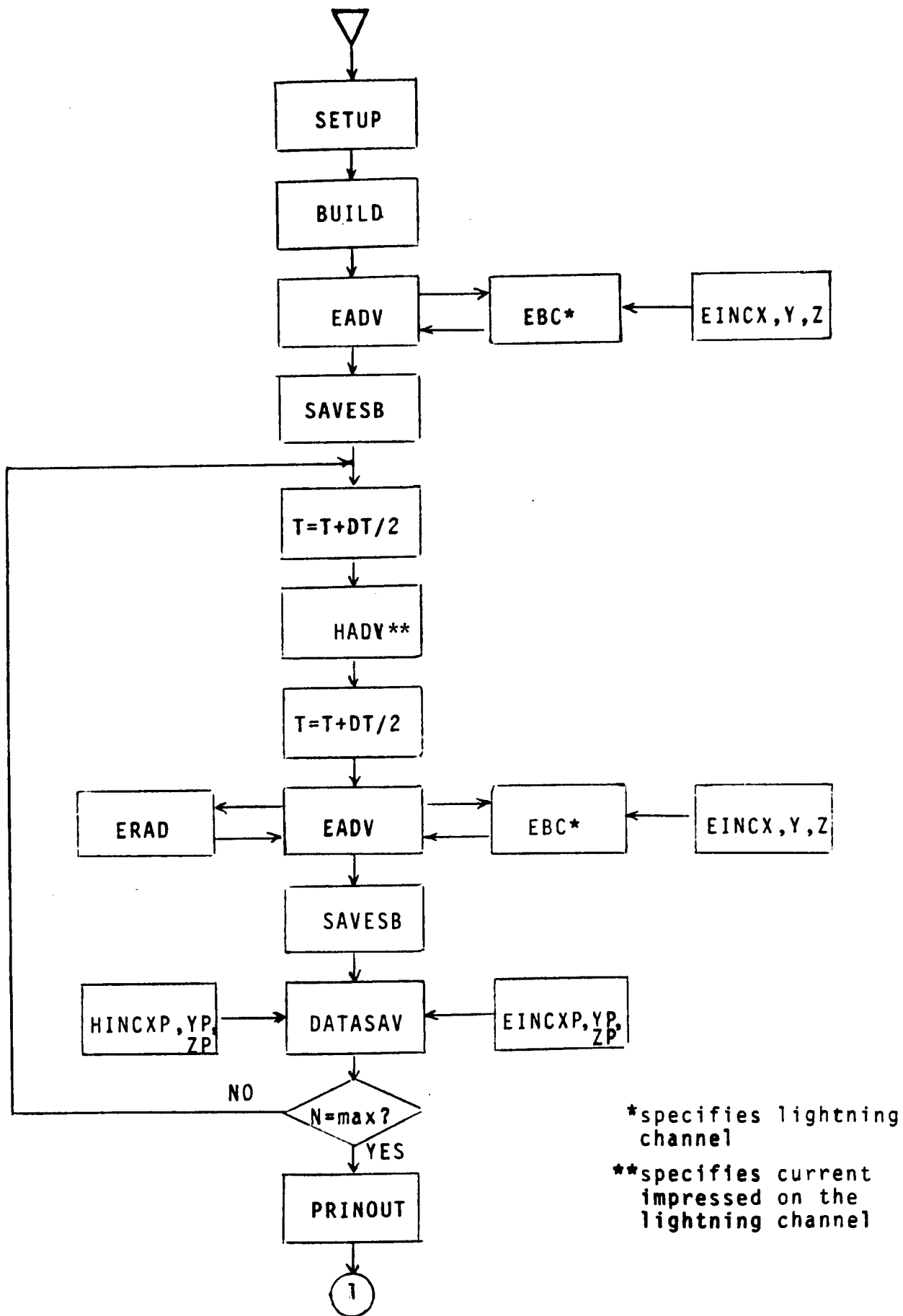


Figure 2b. First Pass, Diffusion Calculation

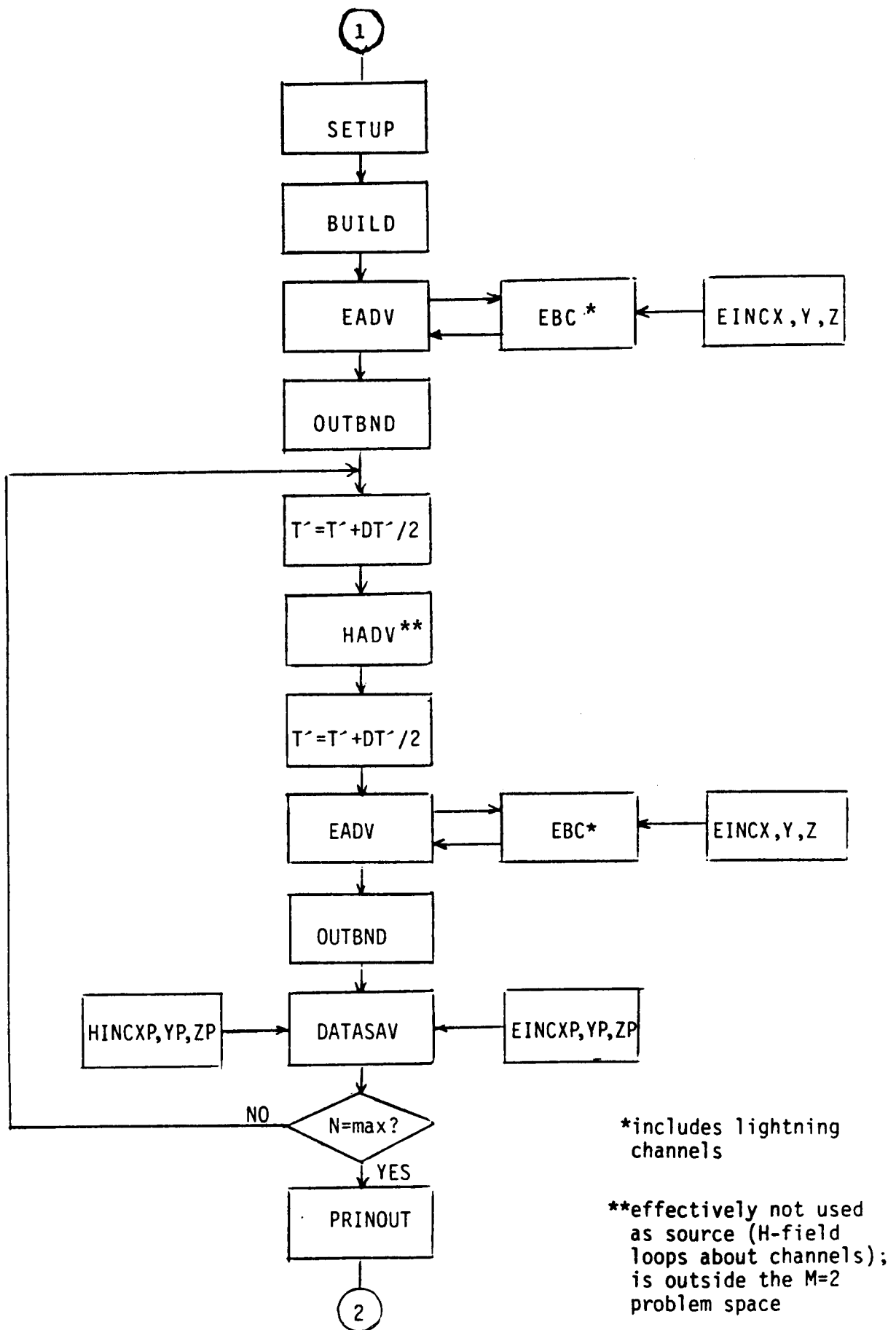


Figure 2c. Second Pass, Expanded Diffusion Calculation

```

PROGRAM DRIVER(INPUT,OUTPUT=OUTKF)

C
COMMON/EFIELD/EX(29,29,29),EY(29,29,29),EZ(29,29,29)
COMMON/HFIELD/HX(29,29,29),HY(29,29,29),HZ(29,29,29)
COMMON/GRID/X(28),Y(28),Z(28),XO(29),YO(29),ZO(29),
1 DX(29),DY(29),DZ(29),DXO(28),DYO(28),DZO(28),
2 DXI(29),DYI(29),DZI(29),DXOI(28),DYOI(28),DZOI(28)
COMMON/EXTRAS/NX,NY,NZ,NX1,NY1,NZ1,N,M,MQ,DT,XMU,EPSO,EPS,NPTLIM,
1 NN,NPTS,LMAX,SIGMA,C,T,PI,EXPFAC,IP,TX,TY,TZ,AMP,ALPHA,BETA,IDLS
COMMON/PERM/MM

C
C LOOP OVER THE DESIRED PERMUTATION OF THE TWO CONFIGURATIONS
C (UNEXPANDED M=1/EXPANDED ,M=2)
C
C SET MM VALUE TO DETERMINE WHICH PERMUTATION
C (UNEXPANDED ONLY,MM=1/EXPANDED ,MM=12)
C
MM=12

C
PRINT 10
10 FORMAT(*MM INPUT ERROR*)
C
100 CONTINUE
C
C UNEXPANDED(M=1) LOOP
C
M=1

C
C GENERATE PROBLEM SPACE AND INTERACTION OBJECT FOR M=1
C
CALL SETUP
CALL BUILD

C
C SET TIME LIMITS AND EVERY IP DATA POINT SAVED
C
TSTART=0.0
PRINT 111
111 FORMAT(*BUILD DONE*)
C
T=DT/2.+TSTART
N=0
CALL EADV
IF(MM.NE.1) CALL SAVESB
PRINT 112
112 FORMAT(*EADV CALLED*)
DO 130 N=1,48

C
C ADVANCE TIME
C
T=T+DT/2.

C
C ADVANCE HFIELD
C
CALL HADV
C

```

```

C      ADVANCE TIME
C
      T=T+DT/2.
C
C      ADVANCE EFIELD
C
      CALL EADV
      IF(MM.NE.1) CALL SAVESB
C
C      STORE FIELDS
C
      IF(MOD(N,IP).EQ.0) CALL DATASAV
C
C
130 CONTINUE
C
      PRINT 150,T,N
150 FORMAT(*1EXIT TIME(M=1)=*E12.3*,AFTER CYCLE*I4)
      CALL PRINOUT
      IF(MM.NE.1) GO TO 160
      GO TO 500
C
160 CONTINUE
C
200 CONTINUE
C
      EXPANDED (M=2) LOOP
C
      M=2
C
      GENERATE PROBLEM SPACE AND INTERACTION OBJECT FOR M=2
C
      CALL SETUP
      CALL BUILD
C
      SET TIME LIMITS AND EVERY IP DATA POINT SAVED
C
      TSTART=0.0
      IP=IP*EXPFAC
C
      T=DT/2.+TSTART
      N=0
      CALL EADV
      DO 230 N=1,192
C      ADVANCE TIME
C
      T=T+DT/2.
C
C      ADVANCE HFIELD
C
      CALL HADV
C
      ADVANCE TIME
C
      T=T+DT/2.

```

```

C
C   ADVANCE EFIELD
C
C   CALL EADV
C
C   STORE FIELDS
C
C   IF(MOD(N,IP).EQ.0) CALL DATASAV
C
C
C 230 CONTINUE
C
C   PRINT 250,T,N
250 FORMAT(*1EXIT TIME(M=2)=*E12.3*,AFTER CYCLE*I4)
C   CALL PRINOUT
C   IF(IM.NE.12) GO TO 260
C   GO TO 500
C
C 260 CONTINUE
C
C 300 CONTINUE
C 500 CONTINUE
C   END

```

3.2 GLOBAL VARIABLES

The global variables for G3DXL are located in labeled common. The nine commons are labeled EFIELD, HFIELD, EXTRAS, GRID, UGRID, RAD, TSITEM, EBS, and OUT. The variables are described in Table 2.

3.3 DATA FILES

Input to Program G3DXL is made through data statements and by manually defining program variables, predominantly in subroutine SETUP. The variables input are identified in the discussion of the subroutines in which they are defined (Section 3.5).

There is one output file for Program G3DXL. This printer output is referred to in the program as OUTPUT and is accessed via formatted print statements. Descriptions of the output data are furnished in the discussions of the subroutines in which they are output (Section 3.5).

3.4 ERROR MESSAGES

The only error message in G3DXL is in the main program DRIVER. The statement "MM INPUT ERROR" is printed if MM is not set to an allowed value.

Table 2. Global Variables

Variable Name	Common	Size	Defining Routines	Other Using Routines	Description
C	EXTRAS	1	SETUP	EINCX, EINCY, EINCZ, EINCXP, EINCYP, EINCZP, HINCXP, HINCYP, HINCZP, DATASAV	Constant = 3×10^8 Speed of light in m/sec
DT	EXTRAS	1	SETUP	Main Program, EADV, HADV DATASAV	Cycle time step (seconds)
DX0, DY0 DZ0, DX, DY, DZ	GRID	28	SETUP	HADV, DATASAV	Individual cell dimensions along x, y, z direction (meters)
DX0I, etc.	GRID	28	SETUP	HADV, DATASAV	Individual cell inverse dimensions along x, y, z, direction (meters)
EPS0	EXTRAS	1	SETUP	EADV, DATASAV	Constant = 8.854×10^{-12} Dielectric constant of free space ($\text{coul}^2/\text{n-m}^2$)
EPS	EXTRAS	1	SETUP	EADV, DATASAV	Dielectric constant of scatterer ($\text{coul}^2/\text{n-m}^2$)
ESTORE	OUT	(500, 30)	DATASAV		Electric field intensity at each time value for each test point

Table 2. (continued)

Variable Name	Common	Size	Defining Routines	Other Using Routines	Description
EX	EFIELD	(29, 29, 29)	ERAD, EADV, EBC	HADV, DATASAV	x component of electric field for each cell in the cell space
EXYD	EBS	(28, 29, 3)	EADV	ERAD	Backstored values of x components of E-field used for calculating far field radiation condition
EXYU	EBS	(28, 29, 3)	EADV	ERAD	
EXZD	EBS	(28, 29, 3)	EADV	ERAD	
EXZU	EBS	(28, 29, 3)	EADV	ERAD	
EY	EFIELD	(29, 29, 29)	ERAD, EADV, EBC	HADV, DATASAV	y component of electric field for each cell in the cell space
EYXD	EBS	(28, 29, 3)	EADV	ERAD	Backstored values of y components of E-field used for calculating far field radiation condition
EYXU	EBS	(28, 29, 3)	EADV	ERAD	

Table 2. (continued)

Variable Name	Common	Size	Defining Routines	Other Using Routines	Description
EYZD	EBS	(28, 29,3)	EADV	ERAD	Backstored values of y components of E-field used for calculating far field radiation condition
EYZU	EBS	(28, 29,3)	EADV	ERAD	
EZ	EBS	(29, 29, 29)	ERAD, EADV, EBC	HADV, DATASAV	z component of E-field for each cell in the cell space
EZXD	EBS	(29, 28,3)	EADV	ERAD	Backstored values of z components of E-field used for calculating far field radiation condition
EZXU	EBS	(29, 28,3)	EADV	ERAD	
EZYD	EBS	(29, 28,3)	EADV	ERAD	
EZYU	EBS	(29, 28,3)	EADV	ERAD	
HSTOR1	OUT	(500, 30)	DATASAV		Component of current density at each time value for each test point

Table 2. (continued)

Variable Name	Common	Size	Defining Routines	Other Using Routines	Description
HSTOR2	OUT	(500, 30)	DATASAV		Component of current density at each time value for each test point
HX	HFIELD	(29, 29, 29)	HADV	EADV, DATASAV	x component of magnetic field for each cell face in the cell space
HY	HFIELD	(29, 29, 29)	HADV	EADV, DATASAV	y component of magnetic field for each cell face in the cell space
HZ	HFIELD	(29, 29, 29)	HADV	EADV, DATASAV	z component of magnetic field for each cell face in the cell space
IP	EXTRAS	1	SETUP	DRIVER	Number of cycles between calls to DATASAV to save data
MQ	EXTRAS	1	SETUP	EADV, DATASAV	Index used to advance back-stored values of E-fields
MU	EXTRAS	1	SETUP	EADV, DATASAV	Constant = $4\pi \times 10^{-7}$; absolute permeability of free space (n/A^2)
N	EXTRAS	1	DRIVER	EADV, ERAD, DATASAV	Counter on the number of cycles that have been computed

Table 2. (continued)

Variable Name	Common	Size	Defining Routines	Other Using Routines	Description
NOPE	TSITEM	(29, 29, 29)	BUILD	EBC	Array that defines the test item. Each cell is assigned a number to indicate if it is an element of the test item
NX	EXTRAS	1	SETUP	BUILD, HADV, ERAD, DATASAV	NX1+1
NX1	EXTRAS	1	SETUP	ERAD, EADV, HADV, DATASAV	Number of cells in the cell space on the x dimension
NY	EXTRAS	1	SETUP	BUILD, HADV, ERAD, DATASAV	NY1+1
NY1	EXTRAS	1	SETUP	ERAD, EADV, HADV, DATASAV	Number of cells in the cell space in the y dimension
NZ	EXTRAS	1	SETUP	BUILD, HADV, ERAD, DATASAV	NZ1+1
NZ1	EXTRAS	1	SETUP	ERAD, EADV, HADV, DATASAV	Number of cells in the cell space in the z dimension
N1	EBS	1	EADV	ERAD	Cycle numbers used to compute L1, L2 and L3 (see Subroutine ERAD)
N2	EBS	1	EADV	ERAD	
N3	EBS	1	EADV	ERAD	

Table 2. (continued)

Variable Name	Common	Size	Defining Routines	Other Using Routines	Descriptions
PI	EXTRAS	1	SETUP	DATASAV	Constant = π
RXY	RAD	(28, 29, 2)	SETUP	ERAD	Distance to the E_x component that is to be extrapolated in the y direction
RXZ	RAD	(28, 29, 2)	SETUP	ERAD	Distance to the E_x component that is to be extrapolated in the z direction
RYX	RAD	(28, 29, 2)	SETUP	ERAD	Distance to the E_y component that is to be extrapolated in the x direction
RYZ	RAD	(29, 28, 2)	SETUP	ERAD	Distance to the E_y component that is to be extrapolated in the z direction
RZX	RAD	(29, 28, 2)	SETUP	ERAD	Distance to the E_z component that is to be extrapolated in the x direction
RZY	RAD	(29, 29, 2)	SETUP	ERAD	Distance to the E_z component that is to be extrapolated in the y direction

Table 2. (continued)

Variable Name	Common	Size	Defining Routines	Other Using Routines	Descriptions
T	EXTRAS	1	Main Program	EINX,EINCY,EINCZ, EINCXP,EINCYP,EINCZP, HINCXP,HINCYP,HINCZP, DATASAV	Time value (seconds)
THXY	RAD	(28, 29,2)	SETUP	ERAD	Angle defined in terms of RXY according to Equation (7)
THXZ	RAD	(28, 29,2)	SETUP	ERAD	Angle defined in terms of RXZ according to Equation (7)
THYX	RAD	(28, 29,2)	SETUP	ERAD	Angle defined in terms of RYX according to Equation (7)
THYZ	RAD	(28, 29,2)	SETUP	ERAD	Angle defined in terms of RYZ according to Equation (7)
THZX	RAD	(28, 29,2)	SETUP	ERAD	Angle defined in terms of RZX according to Equation (7)
THZY	RAD	(28, 29,2)	SETUP	ERAD	Angle defined in terms of RZY according to Equation (7)
TFIT	SETUP	300	SETUP	GFIT	Time coordinates for the simulator field driving function (seconds)

Table 2. (continued)

Variable Name	Common	Size	Defining Routines	Other Using Routines	Description
TSTORE	OUT	500	DATASAV		Time values for output data (seconds)
X	GRID	28	SETUP	EINCX,EADV, DATASAV	x coordinates at centers of cells relative to center of test item (meters)
X0	GRID	29	SETUP	EINCY,EINCZ, DATASAV	x coordinates at ends of cells relative to center of test item (meters)
Y	GRID	28	SETUP	EINCY,EADV, DATASAV	y coordinates at centers of cells relative to center of test item (meters)
Y0	GRID	29	SETUP	EINCX,EINCZ, DATASAV	y coordinates at ends of cells relative to center of test item (meters)
Z	GRID	28	SETUP	EINCZ,EADV, DATASAV	z coordinates at centers of cells relative to center of test item (meters)
Z0	GRID	29	SETUP	EINCX,EINCY, DATASAV	z coordinates at ends of cells relative to center of test item (meters)

3.5 SUBROUTINES

In this section the subroutines that comprise Program G3DXL are presented. A brief description of the function performed by the subroutine is provided along with a subroutine flow chart and a description of those variables local to the subroutine.

- SETUP - Problem space definition, initialization and parameter definitions
- BUILD - Scattering geometry definition
- EADV - Advances free space E-fields
- EBC - Sets electric field boundary conditions, lightning channels, and defines internal wires.
- ERAD - Outer radiation boundary condition
- HADV - Advances free space H-fields
- FIELDS - Defines indirect radiated fields
- SAVESB - Saves subboundary tangential E-fields for expansion technique
- OUTBND - Interpolates SAVESB fields for expanded calculation
- DATASAV - Saves data at selected test point locations
- PRINOUT - Outputs data from DATASAV, extends wire current data in time

3.5.1 Subroutine SETUP

The test configuration and conditions are specified in subroutine SETUP. The gridding for the cell space, the time step and the radiation factors are determined. The variables listed in Table 3 are set in the subroutine. A flow chart for Subroutine SETUP is furnished in Figure 3. Table 4 presents the variables local to the subroutine.

Table 3. Variables Input in Subroutine SETUP

Variable Name	Description
DELX	Length of cell where there is no cell expansion (x dimension)(m)
DELY	Width of cell where there is no cell expansion (z dimension)(m)
DELZ	Height of cell where there is no cell expansion (y dimension)(m)
IDOWN	I index of the first cell that is not expanded in the x direction
IUP	I index of the last cell that is not expanded in the x direction
JDOWN	J index of the first cell that is not expanded in the y direction
JUP	J index of the last cell that is not expanded in the y direction
KDOWN	K index of the first cell that is not expanded in the z direction
KUP	K index of the last cell that is not expanded in the z direction
IST JST KST	Starting point in the unexpanded grid (I,J,K) = (IST,JST,KST), for the expanded grid, (I,J,K) = (0,0,0)

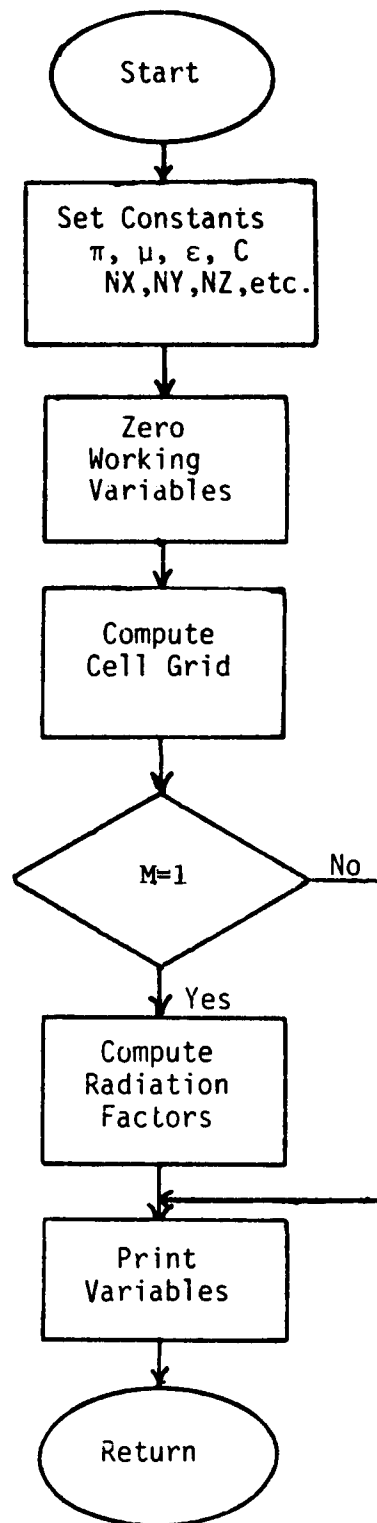


Figure 3. Flow Chart for Subroutine SETUP

Table 4. Local Variables - Subroutine SETUP

Variable Name	Description
DTXI	Minimum cell dimension in x
DTYI	Minimum cell dimension in y
DTZI	Minimum cell dimension in z
XPANX, XPANY XPANZ	Cell expansion coefficient in x, y, z direction
EXPFAC	Expansion factor for M=2,4 loops
IDOWN	See Table 3
IUP	See Table 3
JDOWN	See Table 3
JUP	See Table 3
KDOWN	See Table 3
KUP	See Table 3
Q	Minimum value of QYZ, QZX, QXY, QZY, QXZ, and QYZ
QXY	Minimum value of θ_{xy} 's
QXZ	Minimum value of θ_{xz} 's
QYX	Minimum value of θ_{yx} 's
QYZ	Minimum value of θ_{yz} 's
QZX	Minimum value of θ_{zx} 's
QZY	Minimum value of θ_{zy} 's
R1 R2	Temporary storage for distances in computation of radiation factors

The gridding of the cell space for the unexpanded loops is determined from DELX, DELY, and DELZ (see Table 3) and the expansion coefficients (XPANX, XPANY, and XPANZ). The grid size within the nonexpanding area is simply DELX, DELY, and DELZ in the x, y, and z directions respectively. The majority, if not all, of the test item is usually contained within this area. Outside this area the cell size of each cell as one moves away from the center of the cell space is the previous cell size multiplied by the expansion factor for that coordinate, for example,

$$DXO(I+1) = DXO(I) * EXPANX$$

The time step for the computations is defined as

$$\Delta t = \left\{ \sqrt{\frac{1}{\epsilon \mu}} * \left[\frac{1}{(dx_{\min})^2} + \frac{1}{(dy_{\min})^2} + \frac{1}{(dz_{\min})^2} \right]^{1/2} \right\}^{-1}$$

where dx_{\min} is the minimum cell dimension for the x coordinate direction.

The radiation factors θ_{yx} , θ_{zx} , θ_{xy} , θ_{zy} , θ_{xz} , θ_{yz} , R_{yx} , R_{zx} , R_{xy} , R_{zy} , R_{xz} , and R_{yz} (see Subroutine ERAD) are computed and stored for later use in the computations.

Subroutine SETUP is called from DRIVER, the main program of G3DXL. Variables input and computed in Subroutine SETUP are printed before it returns to the program.

SUBROUTINE SETUP

```

C
  DIMENSION DDXO(210),DDYO(210),DDZO(210)
  COMMON/EFIELD/EX(29,29,29),EY(29,29,29),EZ(29,29,29)
  COMMON/HFIELD/HX(29,29,29),HY(29,29,29),HZ(29,29,29)
  COMMON/GRID/X(28),Y(28),Z(28),XO(29),YO(29),ZO(29),
1  DX(29),DY(29),DZ(29),DXO(28),DYO(28),DZO(28),
2  DXI(29),DYI(29),DZI(29),DXOI(28),DYOI(28),DZOI(28)
  COMMON/UGRID/UX(28),UY(28),UZ(28),UXO(29),UYO(29),UZO(29)
  COMMON/EXTRAS/NX,NY,NZ,NX1,NY1,NZ1,N,M,MQ,DT,XMU,EPSO,EPS,NPTLIM,
1  NN,NPTS,LMAX,SIGMA,C,T,PI,EXPFAC,IP,TX,TY,TZ,AMP,ALPHA,
2  BETA,IDLS
  COMMON/RAD/RYX(28,29,2),RZX(29,28,2),RXY(28,29,2),RZY(29,28,2),
1  RXZ(28,29,2),RYZ(29,28,2),THYX(28,29,2),THZX(29,28,2),
2  THXY(28,29,2),THZY(29,28,2),THXZ(28,29,2),THYZ(29,28,2)
  COMMON/EBS/EYXD(28,29,3),EYXU(28,29,3),EZXD(29,28,3),EZ XU(29,28,3)
1  ,EXYD(28,29,3),EXYU(28,29,3),EZYD(29,28,3),EZYU(29,28,3),
2  EXZD(28,29,3),EXZU(28,29,3),EYZD(29,28,3),EYZU(29,28,3),
3  N1,N2,N3

C
  COMMON/CUMMUL/CUMM(24),LOC(24)
C
  SET CYCLES BETWEEN PRINTS AND RECORD LENGTH
C
  DATA LMAX/250/

C
  SET EXPANSION FACTOR
C
  DATA EXPFAC/4.0/

C
  SET PARAMETERS
C
  DATA PI,EPSO,XMU,C/3.1415926536,8.854E-12,1.256637E-6,3.E8/

C
  SET PROBLEM SPACE DIMENSIONS
C
  DATA NX,NY,NZ,NX1,NY1,NZ1/29,29,29,28,28,28/

C
  SET AIRCRAFT SIGMA AND EPSILON
C
  DATA EPS,SIGMA/3.1E-11,1.0E2/

C
  IDLS=0 FOR RADIATED FIELDS,IDLS=1 FOR DIRECT STRIKES
C
  DATA IDLS/1/

C
  IP=2

C
  FIRST INITIALIZE FIELDS TO ZERO
C
  DO 10 I=1,29
  DO 10 J=1,29
  DO 10 K=1,29

C
  EX(I,J,K)=0.0
  EY(I,J,K)=0.0

```

```

      EZ(I,J,K)=0.0
      HX(I,J,K)=0.0
      HY(I,J,K)=0.0
      HZ(I,J,K)=0.0
10  CONTINUE
C
      DO 20 I=1,28
      DO 20 J=1,29
      DO 20 K=1,3
C
      EXYD(I,J,K)=0.0
      EXYU(I,J,K)=0.0
      EXZD(I,J,K)=0.0
      EXZU(I,J,K)=0.0
      EYXD(I,J,K)=0.0
      EYXU(I,J,K)=0.0
20  CONTINUE
C
      DO 25 I=1,29
      DO 25 J=1,28
      DO 25 K=1,3
      EYZD(I,J,K)=0.0
      EYZU(I,J,K)=0.0
      EZXD(I,J,K)=0.0
      EZXU(I,J,K)=0.0
      EZYD(I,J,K)=0.0
      EZYU(I,J,K)=0.0
25  CONTINUE
C
      N1=0
      N2=0
      N3=0
C
      DO 30 LL=1,24
      CUMM(LL)=0.0
30  CONTINUE
C
      AMP=5.92E4
      ALPHA=4.08E6
      BETA=1.0E8
C
      TX=0.002
      TY=0.005
      TZ=0.005
C
C
C      INPUT GRID FOR M=1
C
C      SET THE UNEXPANDED CELL DIMENSIONS(DE LX,DE LY,DE LZ) IN METERS,
C      EXPANSION FACTORS(XPANX,XPANY,XPANZ) AND THE LIMITS OVER
C      WHICH THE CELLS REMAIN CONSTANT(IUP,IDOWN,JUP,JDOWN,KUP,KDOWN)
      DELX=1.2
      DELY=0.48
      DELZ=0.48

```

```

C      XPANX=1.3
      XPANY=1.15
      XPANZ=1.15
C
      IUP=25
      IDOWN=5
      JUP=25
      JDOWN=5
      KUP=25
      KDOWN=5
C
      DO 100 I=1, IDOWN
      DX0(I)=DELX*(XPANX**(IDOWN-I))
100 CONTINUE
C
      DO 105 I=IDOWN, IUP
      DX0(I)=DELX
105 CONTINUE
C
      DO 110 I=IUP, NX1
      DX0(I)=DELX*(XPANX**(I+1-IUP))
110 CONTINUE
C
      DO 115 J=1, JDOWN
      DY0(J)=DELY*(XPANY**(JDOWN-J))
115 CONTINUE
C
      DO 120 J=JDOWN, JUP
      DY0(J)=DELY
120 CONTINUE
C
      DO 125 J=JUP, NY1
      DY0(J)=DELY*(XPANY**(J+1-JUP))
125 CONTINUE
C
      DO 130 K=1, KDOWN
      DZ0(K)=DELZ*(XPANZ**(KDOWN-K))
130 CONTINUE
C
      DO 135 K=KDOWN, KUP
      DZ0(K)=DELZ
135 CONTINUE
C
      DO 140 K=KUP, NZ1
      DZ0(K)=DELZ*(XPANZ**(K+1-KUP))
140 CONTINUE
C
      IF(M.EQ.1) GO TO 205
C
C      INPUT GRID FOR M=2
C
C
      NX1=EXPFAC*NX1
      NY1=EXPFAC*NY1

```



```

      NZ1=EXPFAC*NZ1
      NX=NX1+1
      NY=NY1+1
      NZ=NZ1+1
C
      IST=7
      JST=9
      KST=12
C
      IDOWN=(EXPFAC*(IDOWN-1))+1
      IUP=(EXPFAC*(IUP-1))+1
      XPANX=XPANX/EXPFAC
C
      JDOWN=(EXPFAC*(JDOWN-1))+1
      JUP=(EXPFAC*(JUP-1))+1
      XPANY=XPANY/EXPFAC
C
      KDOWN=(EXPFAC*(KDOWN-1))+1
      KUP=(EXPFAC*(KUP-1))+1
      XPANZ=XPANZ/EXPFAC
C
      DELX=DELX/EXPFAC
      DELY=DELY/EXPFAC
      DELZ=DELZ/EXPFAC
C
      DO 145 I=1,IDOWN
      DDXO(I)=DELX*(XPANX**(IDOWN-I))
145  CONTINUE
C
      DO 150 I=IDOWN,IUP
      DLXO(I)=DELX
150  CONTINUE
C
      DO 155 I=IUP,NX1
      DDXO(I)=DELX*(XPANX**(I+1-IUP))
155  CONTINUE
C
      DO 160 J=1,JDOWN
      DDYO(J)=DELY*(XPANY**(JDOWN-J))
160  CONTINUE
C
      DO 165 J=JDOWN,JUP
      DDYO(J)=DELY
165  CONTINUE
C
      DO 170 J=JUP,NY1
      DDYO(J)=DELY*(XPANY**(J+1-JUP))
170  CONTINUE
C
      DO 175 K=1,KDOWN
      DDZO(K)=DELZ*(XPANZ**(KDOWN-K))
175  CONTINUE
C
      DO 180 K=KDOWN,KUP
      DDZO(K)=DELZ

```

```

180 CONTINUE
C
  DO 185 K=KUP,NZ1
    DDZO(K)=DELZ*(XPANZ**(K+1-KUP))
185 CONTINUE
C
  IEXPST=(EXPFAC*(IST-1))+1
  IEXPEND=IEXPST+27
C
  DO 190 I=IEXPST,IEXPEND
    DXO(I-IEXPST+1)=DDXO(I)
190 CONTINUE
C
  JEXPST=(EXPFAC*(JST-1))+1
  JEXPEND=JEXPST+27
C
  DO 195 J=JEXPST,JEXPEND
    DYO(J-JEXPST+1)=DDYO(J)
195 CONTINUE
C
  KEXPST=(EXPFAC*(KST-1))+1
  KEXPEND=KEXPST+27
C
  DO 200 K=KEXPST,KEXPEND
    DZO(K-KEXPST+1)=DDZO(K)
200 CONTINUE
C
  NX1=NX1/EXPFAC
  NY1=NY1/EXPFAC
  NZ1=NZ1/EXPFAC
  NX=NX1+1
  NY=NY1+1
  NZ=NZ1+1
C
  IDOWN=((IDOWN-1)/EXPFAC)+1
  IUP=((IUP-1)/EXPFAC)+1
  JDOWN=((JDOWN-1)/EXPFAC)+1
  JUP=((JUP-1)/EXPFAC)+1
  KDOWN=((KDOWN-1)/EXPFAC)+1
  KUP=((KUP-1)/EXPFAC)+1
C
205 CONTINUE
C
  D=0.0
  DO 210 I=1,NX1
    D=D+DXO(I)
210 CONTINUE
C
  DO 215 I=2,NX
    XO(1)=-(D/2.)
    IF(M.EQ.2) XO(1)=UXO(IST)
    XO(I)=XO(I-1)+DXO(I-1)
    IF(M.EQ.2) GO TO 215
    UXO(1)=XO(1)
    UXO(I)=XO(I)

```

```

215 CONTINUE
C
  W=0.0
  DO 220 J=1,NY1
    W=W+DYU(J)
220 CONTINUE
C
  DO 225 J=2,NY
    Y0(1)=- (W/2.)
    IF(M.EQ.2) Y0(1)=UY0(JST)
    Y0(J)=Y0(J-1)+DY0(J-1)
    IF(M.EQ.2) GO TO 225
    UY0(1)=Y0(1)
    UY0(J)=Y0(J)
225 CONTINUE
C
  H=0.0
  DO 230 K=1,NZ1
    H=H+DZO(K)
230 CONTINUE
C
  DO 235 K=2,NZ
    ZO(1)=- (H/2.)
    IF(M.EQ.2) ZO(1)=UZ0(KST)
    ZO(K)=ZO(K-1)+DZO(K-1)
    IF(M.EQ.2) GO TO 235
    UZO(1)=ZO(1)
    UZO(K)=ZO(K)
235 CONTINUE
C
  DO 240 I=1,NX1
    X(I)=(X0(I)+X0(I+1))/2.
    IF(M.EQ.2) GO TO 240
    UX(I)=X(I)
240 CONTINUE
C
  DO 245 J=1,NY1
    Y(J)=(Y0(J)+Y0(J+1))/2.
    IF(M.EQ.2) GO TO 245
    UY(J)=Y(J)
245 CONTINUE
C
  DO 250 K=1,NZ1
    Z(K)=(ZO(K)+ZO(K+1))/2.
    IF(M.EQ.2) GO TO 250
    UZ(K)=Z(K)
250 CONTINUE
C
  DO 260 I=2,NX1
    DX(I)=X(I)-X(I-1)
260 CONTINUE
    DX(1)=DX(2)*XPANX
    DX(29)=DX(28)*XPANX
C
  DO 265 J=2,NY1

```

```

      DY(J)=Y(J)-Y(J-1)
265  CONTINUE
      DY(1)=DY(2)*XPANY
      DY(29)=DY(28)*XPANY
C
      DO 270 K=2,NZ1
      DZ(K)=Z(K)-Z(K-1)
270  CONTINUE
      DZ(1)=DZ(2)*XPANZ
      DZ(29)=DZ(28)*XPANZ
C
      DO 275 I=1,NX1
      DXOI(I)=1./DXO(I)
275  CONTINUE
C
      DO 280 J=1,NY1
      DYOI(J)=1./DYO(J)
280  CONTINUE
C
      DO 285 K=1,NZ1
      DZOI(K)=1./DZO(K)
285  CONTINUE
C
      DO 290 I=1,NX
      DXI(I)=1./DX(I)
290  CONTINUE
C
      DO 295 J=1,NY
      DYOI(J)=1./DY(J)
295  CONTINUE
C
      DO 300 K=1,NZ
      DZI(K)=1./DZ(K)
300  CONTINUE
C
      DTXI=C/DXO(NX1/2)
      DTYI=C/DYO(NY1/2)
      DTZI=C/DZO(NZ1/2)
      DT=1./SQRT(DTXI**2+DTYI**2+DTZI**2)
C
      IF(M.EQ.2) GO TO 199
C
C      EYX RADIATION FACTORS
C
      QYX = 1.
      DO 61 K=1,NZ
      DO 61 J=1,NY1
      X1 = XO(1)**2
      Y2 = Y(J)**2
      Z2 = ZO(K)**2
      R1 = SQRT(X1 + Y2 + Z2)
      X2 = XO(2)**2
      R2 = SQRT(X2 + Y2 + Z2)
      THYX(J,K,1) = 1. - (R1-R2)/(C*DT)
      IF(THYX(J,K,1).LT.QYX) QYX = THYX(J,K,1)
      RYX(J,K,1) = R2/R1

```

```

X1 = X0(NX)**2
R1 = SQRT(X1 + Y2 + Z2)
X2 = X0(NX1)**2
R2 = SQRT(X2 + Y2 + Z2)
THYX(J,K,2) = 1. - (R1-R2)/(C*DT)
IF(THYX(J,K,2).LT.QYX) QYX = THYX(J,K,2)
RYX(J,K,2) = R2/R1
61 CONTINUE

C
C      EZX RADIATION FACTORS
C
QZX = 1.
DO 62 K=1,NZ1
DO 62 J=1,NY
X1 = X0(1)**2
Y2 = Y0(J)**2
Z2 = Z(K)**2
R1 = SQRT(X1 + Y2 + Z2)
X2 = X0(2)**2
R2 = SQRT(X2 + Y2 + Z2)
THZX(J,K,1) = 1. - (R1-R2)/(C*DT)
IF(THZX(J,K,1).LT.QZX) QZX = THZX(J,K,1)
RZX(J,K,1) = R2/R1
X1 = X0(NX)**2
R1 = SQRT(X1 + Y2 + Z2)
X2 = X0(NX1)**2
R2 = SQRT(X2 + Y2 + Z2)
THZX(J,K,2) = 1. - (R1-R2)/(C*DT)
IF(THZX(J,K,2).LT.QZX) QZX = THZX(J,K,2)
RZX(J,K,2) = R2/R1
62 CONTINUE

C
C      EXY RADIATION FACTORS
C
QXY = 1.
DO 63 K=1,NZ
DO 63 I=1,NX1
X2 = X(I)**2
Y1 = Y0(1)**2
Z2 = Z0(K)**2
R1 = SQRT(X2 + Y1 + Z2)
Y2 = Y0(2)**2
R2 = SQRT(X2 + Y2 + Z2)
THXY(I,K,1) = 1. - (R1-R2)/(C*DT)
IF(THXY(I,K,1).LT.QXY) QXY = THXY(I,K,1)
RXY(I,K,1) = R2/R1
Y1 = Y0(NY)**2
R1 = SQRT(X2 + Y1 + Z2)
Y2 = Y0(NY1)**2
R2 = SQRT(X2 + Y2 + Z2)
THXY(I,K,2) = 1. - (R1-R2)/(C*DT)
IF(THXY(I,K,2).LT.QXY) QXY = THXY(I,K,2)
RXY(I,K,2) = R2/R1
63 CONTINUE

C

```

C
C

EZY RADIATION FACTORS

```
QZY = 1.
DO 64 K=1,NZ1
DO 64 I=1,NX
X2 = XO(I)**2
Y1 = YO(1)**2
Z2 = Z(K)**2
R1 = SQRT(X2 + Y1 + Z2)
Y2 = YO(2)**2
R2 = SQRT(X2 + Y2 + Z2)
THZY(I,K,1) = 1. - (R1-R2)/(C*DT)
IF(THZY(I,K,1).LT.QZY) QZY = THZY(I,K,1)
RZY(I,K,1) = R2/R1
Y1 = YO(NY)**2
R1 = SQRT(X2 + Y1 + Z2)
Y2 = YO(NY1)**2
R2 = SQRT(X2 + Y2 + Z2)
THZY(I,K,2) = 1. - (R1-R2)/(C*DT)
IF(THZY(I,K,2).LT.QZY) QYZ=THYZ(I,K,2)
RZY(I,K,2)=R2/R1
64 CONTINUE
```

C
C

EXZ RADIATION FACTORS

```
QXZ = 1.
DO 65 I=1,NX1
DO 65 J=1,NY
X2 = X(I)**2
Y2 = YO(J)**2
Z1 = ZO(1)**2
R1 = SQRT(X2 + Y2 + Z1)
Z2 = ZO(2)**2
R2 = SQRT(X2 + Y2 + Z2)
THXZ(I,J,1) = 1. - (R1-R2)/(C*DT)
IF(THXZ(I,J,1).LT.QXZ) QXZ = THXZ(I,J,1)
RXZ(I,J,1) = R2/R1
Z1 = ZO(NZ)**2
R1 = SQRT(X2 + Y2 + Z1)
Z2 = ZO(NZ1)**2
R2 = SQRT(X2 + Y2 + Z2)
THXZ(I,J,2) = 1. - (R1-R2)/(C*DT)
IF(THXZ(I,J,2).LT.QXZ) QXZ = THXZ(I,J,2)
RXZ(I,J,2) = R2/R1
65 CONTINUE
```

C
C

EYZ RADIATION FACTORS

```
QYZ = 1.
DO 66 J=1,NY1
DO 66 I=1,NX
X2 = XO(I)**2
Y2 = Y(J)**2
Z1 = ZO(1)**2
R1 = SQRT(X2 + Y2 + Z1)
Z2 = ZO(2)**2
R2 = SQRT(X2 + Y2 + Z2)
```

```

        THYZ(I,J,1)=1.-(R1-R2)/(C*DT)
        IF(THYZ(I,J,1).LT.QYZ) QYZ=THYZ(I,J,1)
        RYZ(I,J,1)=R2/R1
        Z1=Z0(NZ)**2
        R1=SQRT(X2+Y2+Z1)
        Z2=Z0(NZ1)**2
        R2=SQRT(X2+Y2+Z2)
        THYZ(I,J,2) = 1. - (R1-R2)/(C*DT)
        IF(THYZ(I,J,2).LT.QYZ) QYZ = THYZ(I,J,2)
        RYZ(I,J,2) = R2/R1
66      CONTINUE
        Q = AMIN1(QYX,QZX,QXY,QZY,QXZ,QYZ)
        MQ = -Q/2.

C
C          PRINT VARIABLES
C
199     PRINT 201
        PRINT 202, DT, EPS,SIGMA
        PRINT 99
        PRINT 104, NX, NX1, NY, NY1, NZ, NZ1
        PRINT 101
        PRINT 98, (X0(I), I = 1,NX)
        PRINT 102
        PRINT 98, (Y0(J), J = 1,NY)
        PRINT 107
        PRINT 98, (DX0(I), I = 1,NX1)
        PRINT 103
        PRINT 98, (Z0(K), K = 1,NZ)
        PRINT 106
        PRINT 98, QYX,QZX,QXY,QZY,QXZ,QYZ,Q

C
C          FORMAT STATEMENTS
C
        98 FORMAT (10F10.4)
        101 FORMAT (*OXO GRID*)
        102 FORMAT (*OYO GRID*)
        103 FORMAT (*OZO GRID*)
        104 FORMAT (20I5)
        99 FORMAT(5H0 NX,2X,*NX1 NY NY1 NZ NZ1 MQ*)
        106 FORMAT(1H0,5X,*QYX*,6X,*QZX*,6X,*QXY*,6X,*QZY*,6X,*QXZ*,6X,*QYZ*,
1          7X,*Q*)
        107 FORMAT (*ODXO GRID*)
        202 FORMAT (3E12.4)
        201 FORMAT (*1DT,EPS,SIGMA*)
        390 FORMAT(2F10.3,2E12.4)
        400 FORMAT(3F10.0)
        440 FORMAT(6I5)
        RETURN
        END
C

```

3.5.2 Subroutine BUILD

The NOPE array, which defines the test time, is filled in Subroutine BUILD. Each cell in the cell space is identified by three indices (I,J,K) (see Figure 1). The entire array is initialized to zero. For perfect conductors, if the I,J,Kth cell is a three-dimensional cell in the test time, NOPE(I,J,K) is set equal to four (4). If the YZ plane adjacent to vertex I,J,K of the cell is a two-dimensional piece of the test time, NOPE(I,J,K) is set equal to one (1). For the XZ plane, NOPE(I,J,K) is set to two (2) and to three (3) if the XY plane is a two-dimensional part of the test item. Where planes meet NOPE is set to 12 (YZ and XZ planes), 13 (YZ and XY planes), 23 (XZ and XY planes) and 123 (YZ, XZ and XY planes). When an imperfect conductor is modeled $4 \rightarrow 8$ and $1,2,3 \rightarrow 5,6,7$. The values of the NOPE array for each XZ plane cut of the test item are printed and can be used to check the test item structure. This routine is replaced in its entirety whenever a computation on a new structure is required.

The version of Subroutine BUILD included here builds the F-106B aircraft. A flow chart for the routine is given in Figure 4. Subroutine BUILD is called by the main program DRIVER.

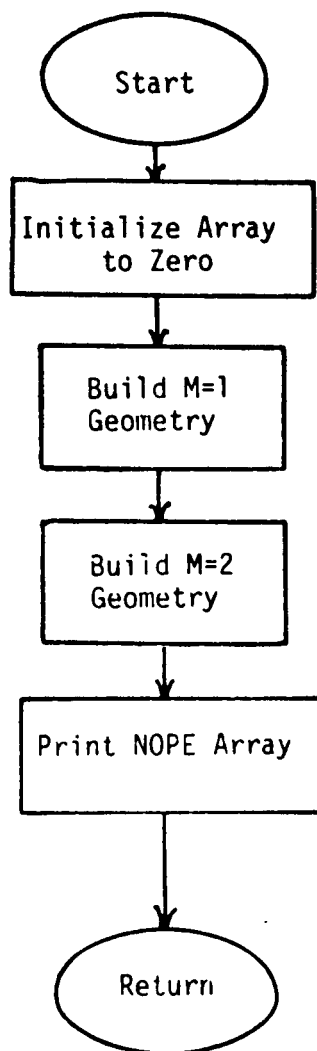


Figure 4. Flow Chart for Subroutine BUILD

```

SUBROUTINE BUILD
C
COMMON/EXTRAS/NX,NY,NZ,NX1,NY1,NZ1,N,M,MQ,DT,XMU,EPSO,EPS,NPTLIM,
1 NN,NPTS,LMAX,SIGMA,C,T,PI,EXPFAC,IP,TX,TY,TZ,AMP,ALPHA,BETA,IDLS
COMMON/TSITEM/NOPE(29,29,29)
C
DO 100 I=1,NX
DO 100 J=1,NY
DO 100 K=1,NZ
NOPE(I,J,K)=0
100 CONTINUE
C
C CYLINDER TEST CASE FOR CODE OPERATION CHECK
C
TEST=0
IF(TEST.EQ.0) GO TO 8888
C
C BUILD M=1 GEOMETRY
C IF(M.NE.1) GO TO 200
DO 110 I=5,24
DO 110 J=13,16
DO 110 K=13,16
NOPE(I,J,K)=8
110 CONTINUE
GO TO 700
1100 CONTINUE
DO 1101 I=5,25,20
DO 1101 J=13,16
DO 1101 K=13,16
NOPE(I,J,K)=5
1101 CONTINUE
DO 1102 I=5,24
DO 1102 J=13,17,4
DO 1102 K=13,16
IF(NOPE(I,J,K).EQ.1) GO TO 1111
NOPE(I,J,K)=6
GO TO 1102
1111 NOPE(I,J,K)=56
1102 CONTINUE
DO 1103 I=5,24
DO 1103 J=13,16
DO 1103 K=13,17,4
IF(NOPE(I,J,K).EQ.1) GO TO 1112
IF(NOPE(I,J,K).EQ.2) GO TO 1113
IF(NOPE(I,J,K).EQ.12) GO TO 1114
NOPE(I,J,K)=7
GO TO 1103
1112 NOPE(I,J,K)=57
GO TO 1103
1113 NOPE(I,J,K)=67
GO TO 1103
1114 NOPE(I,J,K)=567
1103 CONTINUE
GO TO 700
C

```

```

C      BUILD M=2 GEOMETRY
200  CONTINUE
      IF(M.NE.2) GO TO 300
      DO 120 I=1,28
      DO 120 J=5,20
      DO 120 K=5,20
      NOPE(I,J,K)=8
120  CONTINUE
C
      GO TO 700
C
8888  CONTINUE
C
C      BUILD M=1 F 106 B GEOMETRY
C
      IF(M.NE.1) GO TO 1200
C
C      FUSELAGE
C
      DO 10 I=8,20
      DO 10 J=11,12
      DO 10 K=13,16
      NOPE(I,J,K)=4
10  CONTINUE
C
      DO 11 I=11,22
      DO 11 J=13,14
      DO 11 K=13,16
      NOPE(I,J,K)=4
11  CONTINUE
C
      NOPE(9,13,14)=4
      NOPE(9,13,15)=4
C
      NOPE(10,13,13)=4
      NOPE(10,13,14)=4
      NOPE(10,13,15)=4
      NOPE(10,13,16)=4
C
      NOPE(10,14,14)=4
      NOPE(10,14,15)=4
C
      DO 12 I=21,22
          J=12
      DO 12 K=13,16
      NOPE(I,J,K)=4
12  CONTINUE
C
      DO 13 I=13,15
      DO 13 J=12,13
      DO 13 K=12,17,5
      NOPE(I,J,K)=4
13  CONTINUE
C
C      WINGS

```

```

C      DO 14 I=14,20
          J=12
      DO 14 K=9,20
          NOPE(I,J,K)=4
14 CONTINUE
C
      NOPE(14,12,10)=0
      NOPE(14,12,19)=0
      NOPE(14,12,9)=0
      NOPE(14,12,20)=0
      NOPE(15,12,9)=0
      NOPE(15,12,20)=0
C
      DO 15 I=18,20
          J=12
      DO 15 K=7,22
          NOPE(I,J,K)=4
15 CONTINUE
C
      NOPE(18,12,7)=0
      NOPE(18,12,22)=0
C
C      STABILIZER
C
      DO 16 I=19,20
      DO 16 J=15,17
          K=15
      NOPE(I,J,K)=3
16 CONTINUE
C
      NOPE(19,17,15)=0
C
      DO 17 I=21,22
      DO 17 J=15,21
          K=15
      NOPE(I,J,K)=3
17 CONTINUE
C
      NOPE(21,20,15)=0
      NOPE(21,21,15)=0
C
      GO TO 700
C
1200 CONTINUE
C
C      BUILD M=2 GEOMETRY
C
      IF(M.NE.2) GO TO 1300
C
C      BUILD SOLID PORTION FIRST
C
C      TREAT SOLID PORTION ON SUBBOUNDARY
C
      I=28

```

```

        DO 20 J=9,24
        DO 20 K=5,20
        NOPE(I,J,K)=4
20 CONTINUE
C
        DO 21 J=13,20
        DO 21 K=1,24
        NOPE(I,J,K)=4
21 CONTINUE
C
C      TREAT SOLID UPPER BODY
C
        DO 22 I=17,27
        DO 22 J=15,24
        DO 22 K=6,19
        NOPE(I,J,K)=4
22 CONTINUE
C
        DO 23 I=13,16
        DO 23 J=13,20
        DO 23 K=6,19
        NOPE(I,J,K)=4
23 CONTINUE
C
        DO 24 I=13,16
        DO 24 J=21,24
        DO 24 K=7,18
        NOPE(I,J,K)=4
24 CONTINUE
C
        DO 25 I=10,12
        DO 25 J=13,18
        DO 25 K=6,19
        NOPE(I,J,K)=4
25 CONTINUE
C
        I=12
        DO 26 J=19,23
        DO 26 K=8,17
        NOPE(I,J,K)=4
26 CONTINUE
C
        I=11
        DO 27 J=19,22
        DO 27 K=9,16
        NOPE(I,J,K)=4
27 CONTINUE
C
        I=10
        DO 28 J=19,20
        DO 28 K=10,15
        NOPE(I,J,K)=4
28 CONTINUE
C
        I=9

```

```

DO 29 J=18,19
DO 29 K=11,14
NOPE(I,J,K)=4
29 CONTINUE
C
DO 30 I=6,9
J=17
DO 30 K=7,18
NOPE(I,J,K)=4
30 CONTINUE
C
DO 31 I=4,5
J=16
DO 31 K=8,17
NOPE(I,J,K)=4
31 CONTINUE
C
C TREAT Y-Z THIN PLANES(NOPE(I,J,K)=1)
C
C BULKHEAD B1
C
I=4
DO 32 J=9,15
DO 32 K=8,17
NOPE(I,J,K)=1
32 CONTINUE
C
C HOLES IN B1
C
DO 33 J=9,14
DO 33 K=12,13
NOPE(I,J,K)=0
33 CONTINUE
C
C BULKHEAD B2
C
I=6
DO 34 J=9,16
DO 34 K=7,18
NOPE(I,J,K)=1
34 CONTINUE
C
C HOLES IN B2
C
DO 35 J=9,10
DO 35 K=14,17
NOPE(I,J,K)=0
35 CONTINUE
C
C BULKHEAD B4
C
I=30
DO 36 J=9,12
DO 36 K=6,19
NOPE(I,J,K)=1

```

```

36 CONTINUE
C
C   HOLES IN B4
C
      DO 37 J=9,10
      DO 37 K=17,18
      NOPE(I,J,K)=0
37 CONTINUE
C
C   BULKHEAD B6
C
      I=16
      DO 38 J=9,14
      DO 38 K=6,19
      NOPE(I,J,K)=1
38 CONTINUE
C
C   HOLES IN B6
C
      DO 39 J=9,10
      DO 39 K=18,19
      NOPE(I,J,K)=0
39 CONTINUE
C
C   BULKHEAD B8
C
      I=20
      DO 40 J=9,14
      DO 40 K=6,19
      NOPE(I,J,K)=1
40 CONTINUE
C
C   HOLES IN B8
C
      DO 41 J=13,14
      DO 41 K=18,19
      NOPE(I,J,K)=0
41 CONTINUE
C
C   TREAT X-Z THIN PLANES(NOPE(I,J,K)=2)
C
      DO 42 I=4,27
      J=9
      DO 42 K=6,19
      IF(NOPE(I,J,K).EQ.1) GO TO 420
      NOPE(I,J,K)=2
      GO TO 42
420 NOPE(I,J,K)=12
42 CONTINUE
C
      DO 43 I=4,9
      DO 43 K=6,19,13
      NOPE(I,J,K)=0
43 CONTINUE
C

```

```

DO 44 I=4,5
DO 44 K=7,18,11
NOPE(I,J,K)=0
44 CONTINUE
C
C   TREAT X-Y THIN PLANES(NOPE(I,J,K)=3)
C
DO 45 I=10,15
DO 45 J=9,12
DO 45 K=6,20,14
IF(NOPE(I,J,K).EQ.1) GO TO 450
IF(NOPE(I,J,K).EQ.2) GO TO 451
IF(NOPE(I,J,K).EQ.12) GO TO 452
NOPE(I,J,K)=3
GO TO 45
450 NOPE(I,J,K)=13
GO TO 45
451 NOPE(I,J,K)=23
GO TO 45
452 NOPE(I,J,K)=123
45 CONTINUE
C
DO 46 I=16,27
DO 46 J=9,14
DO 46 K=6,20,14
IF(NOPE(I,J,K).EQ.1) GO TO 460
IF(NOPE(I,J,K).EQ.2) GO TO 461
IF(NOPE(I,J,K).EQ.12) GO TO 462
NOPE(I,J,K)=3
GO TO 46
460 NOPE(I,J,K)=13
GO TO 46
461 NOPE(I,J,K)=23
GO TO 46
462 NOPE(I,J,K)=123
46 CONTINUE
C
DO 47 I=6,9
DO 47 J=9,16
DO 47 K=7,19,12
IF(NOPE(I,J,K).EQ.1) GO TO 470
IF(NOPE(I,J,K).EQ.2) GO TO 471
IF(NOPE(I,J,K).EQ.12) GO TO 472
NOPE(I,J,K)=3
GO TO 47
470 NOPE(I,J,K)=13
GO TO 47
471 NOPE(I,J,K)=23
GO TO 47
472 NOPE(I,J,K)=123
47 CONTINUE
C
DO 48 I=4,5
DO 48 J=9,15
DO 48 K=8,18,10

```



```

        IF(NOPE(I,J,K).EQ.1) GO TO 480
        IF(NOPE(I,J,K).EQ.2) GO TO 481
        IF(NOPE(I,J,K).EQ.12) GO TO 482
        NOPE(I,J,K)=3
        GO TO 48
480 NOPE(I,J,K)=13
        GO TO 48
481 NOPE(I,J,K)=23
        GO TO 48
482 NOPE(I,J,K)=123
        48 CONTINUE
C
        GO TO 700
C
401 FORMAT(1H1,/,/,5X,7HNOPE(I,I2,3H,K),/)
402 FORMAT(20I4)
700 CONTINUE
        RETURN
        END

```

3.5.3 Subroutine EADV

In subroutine EADV, the equation

$$\vec{\nabla} \times \vec{H} = \epsilon \frac{\partial \vec{E}}{\partial t} + \sigma \vec{E} \quad (1)$$

is applied to the scattered fields and is used to find \vec{E} from \vec{E} and \vec{H} at earlier times. This "time stepping" solution is found for each component of E, i.e., Equation (1) is written explicitly for each component as

$$\epsilon \frac{\partial E_x}{\partial t} + \sigma E_x = \frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z}$$

$$\epsilon \frac{\partial E_y}{\partial t} + \sigma E_y = \frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x}$$

$$\epsilon \frac{\partial E_z}{\partial t} + \sigma E_z = \frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y}$$

These equations are cast in their finite-difference form and solved for E at the latest time. Thus

$$E_x^{n+1/2}(I,J,K) = E_x^{n-1/2}(I,J,K) \quad (2)$$

$$+ (\Delta t/\epsilon) \frac{H_z^n(I,J,K) - H_z^n(I,J-1,K)}{Y(J) - Y(J-1)}$$

$$- \frac{H_y^n(I,J,K) - H_y^n(I,J,K-1)}{Z(K) - Z(K-1)}$$

$$E_y^{n+1/2}(I,J,K) = E_y^{n-1/2}(I,J,K) + (\Delta t/\epsilon) \quad (3)$$

$$\left(\frac{H_x^n(I,J,K) - H_x^n(I,J,K-1)}{Z(K) - Z(K-1)} - \frac{H_z^n(I,J,K) - H_z^n(I-1,J,K)}{X(I) - X(I-1)} \right)$$

$$E_z^{n+1/2}(I,J,K) = E_z^{n-1/2}(I,J,K) + (\Delta t/\epsilon) \quad (4)$$

$$\left(\frac{H_y^n(I,J,K) - H_y^n(I-1,J,K)}{X(I) - X(I-1)} - \frac{H_x^n(I,J,K) - H_x^n(I,J-1,K)}{Y(J) - Y(J-1)} \right)$$

EADV is called every program cycle by the main program, as is the Subroutine HADV for advancing the H-fields. It uses the prior values of the fields to calculate the new values. Initially, these prior values are zero. The first nonzero values are those values for E and H that imposed at the scatterer's boundary. These values are accessed by a call to EBC after the Equations (2), (3), (4) have been advanced in EADV. Thus, on the next program cycle these values will be available and nonzero values of the fields off the boundary will start appearing when Equations (2), (3), and (4) are advanced again. Since EBC is always called after the fields are advanced, the proper boundary conditions are maintained.

Additionally, EADV stores the E-field components one cell in from each face of the outer boundary that are used in the calculation of the E-field component values on the outer boundary. This calculation, based on the radiation boundary condition [2], is performed in Subroutine ERAD.

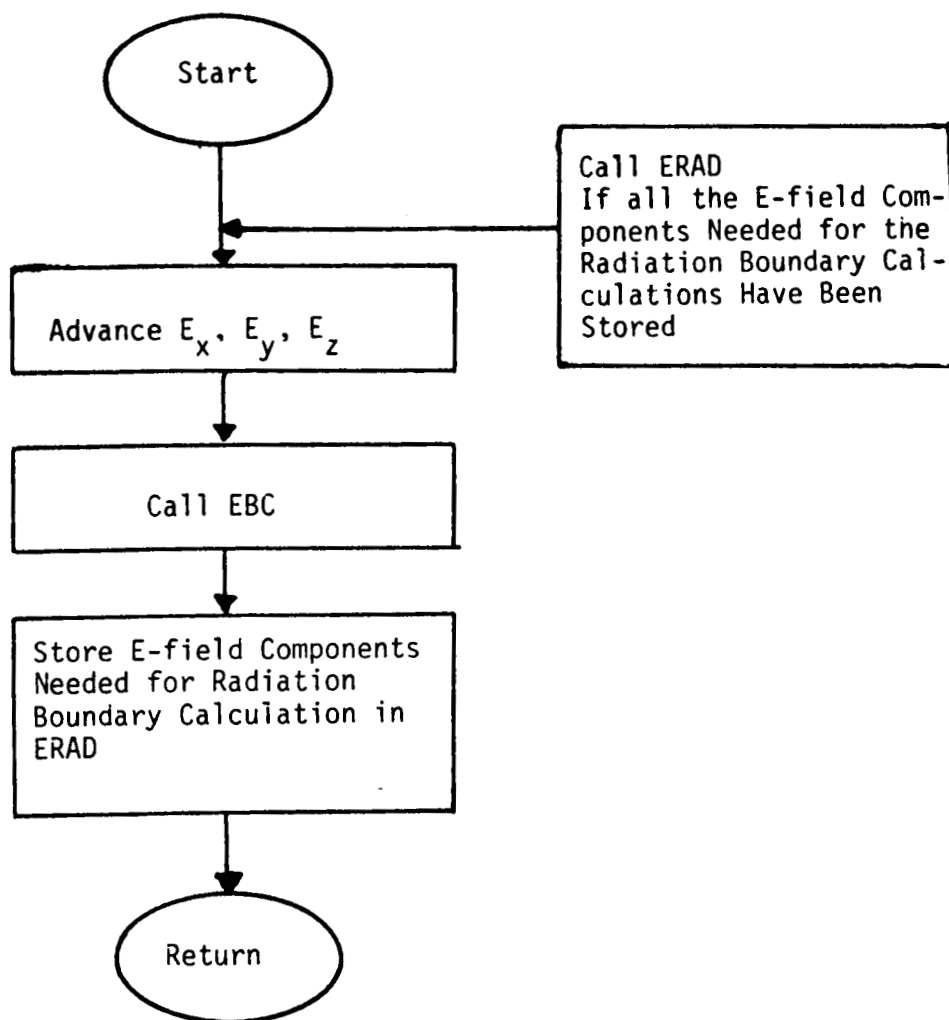


Figure 5. Flow Chart for Subroutine EADV

SUBROUTINE EADV

C

```
COMMON/EFIELD/EX(29,29,29),EY(29,29,29),EZ(29,29,29)
COMMON/HFIELD/HX(29,29,29),HY(29,29,29),HZ(29,29,29)
COMMON/EXTRAS/NX,NY,NZ,NX1,NY1,NZ1,N,M,MQ,DT,XMU,EPSO,EPS,NPTLIM,
1 NN,NPTS,LMAX,SIGMA,C,T,PI,EXPFAC,IP,TX,TY,TZ,AMP,ALPHA,BETA,IDLS
COMMON/GRID/X(28),Y(28),Z(28),XO(29),YO(29),ZO(29),
1 DX(29),DY(29),DZ(29),DXO(28),DYO(28),DZO(28),
2 DXI(29),DYI(29),DZI(29),DXOI(28),LYOI(28),LZOI(28)
COMMON/EBS/EYXD(28,29,3),EYXU(28,29,3),EZXD(29,28,3),EZ XU(29,28,3)
1 ,EXYD(28,29,3),EXYU(28,29,3),EZYD(29,28,3),EZYU(29,28,3),
2 EXZD(28,29,3),EXZU(28,29,3),EYZD(29,28,3),EYZU(29,28,3),
3 N1,N2,N3
COMMON/TSITEM/NOPE(29,29,29)
```

C

DTE=DT/EPSO

C

C

ADVANCE EX

C

```
DO 1 I = 1,NX1
DO 1 J = 2,NY1
DO 1 K = 2,NZ1
IF(NOPE(I,J,K) .NE.0.AND.NOPE(I,J,K) .NE.1.AND.NOPE(I,J,K) .NE.5)
1 GO TO 1
EX(I,J,K)=EX(I,J,K)+DTE*((HZ(I,J,K)-HZ(I,J-1,K))*DYI(J)
1 -(HY(I,J,K)-HY(I,J,K-1))*DZI(K))
1 CONTINUE
```

C

C

ADVANCE EY

C

```
DO 2 I = 2,NX1
DO 2 J = 1,NY1
DO 2 K = 2,NZ1
IF(NOPE(I,J,K) .NE.0.AND.NOPE(I,J,K) .NE.2.AND.NOPE(I,J,K) .NE.6)
1 GO TO 2
EY(I,J,K)=EY(I,J,K)+DTE*((HX(I,J,K)-HX(I,J,K-1))*DZI(K)
1 -(HZ(I,J,K)-HZ(I-1,J,K))*DXI(I))
2 CONTINUE
```

C

C

ADVANCE EZ

C

```
DO 3 I = 2,NX1
DO 3 J = 2,NY1
DO 3 K = 1,NZ1
IF(NOPE(I,J,K) .NE.0.AND.NOPE(I,J,K) .NE.3.AND.NOPE(I,J,K) .NE.7)
1 GO TO 3
EZ(I,J,K)=EZ(I,J,K)+DTE*((HY(I,J,K)-HY(I-1,J,K))*DXI(I)
1 -(HX(I,J,K)-HX(I,J-1,K))*DYI(J))
3 CONTINUE
```

C

```
IF (M.EQ.2) GO TO 10
IF(N2.NE.N3) CALL ERAD
10 CONTINUE
CALL EBC
IF (M.EQ.2) CALL OUTBND
```

```

      IF (M.EQ.2) RETURN
      IF (MOD(N,MQ).NE.0) RETURN
      N3=N2
      N2=N1
      N1=N
C
C
C   ADVANCE EXY
C
      DO 11 I = 1,NX1
      DO 11 K = 2,NZ1
      EXYD(I,K,3) = EXYD(I,K,2)
      EXYD(I,K,2) = EXYD(I,K,1)
      EXYD(I,K,1) = EX(I,2,K)
      EXYU(I,K,3) = EXYU(I,K,2)
      EXYU(I,K,2) = EXYU(I,K,1)
      EXYU(I,K,1) = EX(I,NY1,K)
11 CONTINUE
C
C   ADVANCE EXZ
C
      DO 111 I = 1,NX1
      DO 111 J = 2,NY1
      EXZD(I,J,3) = EXZD(I,J,2)
      EXZD(I,J,2) = EXZD(I,J,1)
      EXZD(I,J,1) = EX(I,J,2)
      EXZU(I,J,3) = EXZU(I,J,2)
      EXZU(I,J,2) = EXZU(I,J,1)
      EXZU(I,J,1) = EX(I,J,NZ1)
111 CONTINUE
C
C   ADVANCE EYX
C
      DO 22 J = 1,NY1
      DO 22 K = 2,NZ1
      EYXD(J,K,3) = EYXD(J,K,2)
      EYXD(J,K,2) = EYXD(J,K,1)
      EYXD(J,K,1) = EY(2,J,K)
      EYXU(J,K,3) = EYXU(J,K,2)
      EYXU(J,K,2) = EYXU(J,K,1)
      EYXU(J,K,1) = EY(NX1,J,K)
22 CONTINUE
C
C   ADVANCE EYZ
C
      DO 222 I = 2,NX1
      DO 222 J = 1,NY1
      EYZD(I,J,3) = EYZD(I,J,2)
      EYZD(I,J,2) = EYZD(I,J,1)
      EYZD(I,J,1) = EY(I,J,2)
      EYZU(I,J,3) = EYZU(I,J,2)
      EYZU(I,J,2) = EYZU(I,J,1)
      EYZU(I,J,1) = EY(I,J,NZ1)
222 CONTINUE
C

```

```

C   ADVANCE EZX
C
      DO 33 J = 2,NY1
      DO 33 K = 1,NZ1
        EZXD(J,K,3) = EZXD(J,K,2)
        EZXD(J,K,2) = EZXD(J,K,1)
        EZXD(J,K,1) = EZ(2,J,K)
        EZXU(J,K,3) = EZXU(J,K,2)
        EZXU(J,K,2) = EZXU(J,K,1)
        EZXU(J,K,1) = EZ(NX1,J,K)
      33 CONTINUE

```

```

C
C   ADVANCE EZY
C
      DO 333 I = 2,NX1
      DO 333 K = 1,NZ1
        EYZD(I,K,3) = EYZD(I,K,2)
        EYZD(I,K,2) = EYZD(I,K,1)
        EYZD(I,K,1) = EZ(I,2,K)
        EYZU(I,K,3) = EYZU(I,K,2)
        EYZU(I,K,2) = EYZU(I,K,1)
        EYZU(I,K,1) = EZ(I,NY1,K)
      333 CONTINUE

```

```

C
      RETURN
      END

```

3.5.4 Subroutine EBC

Subroutine EBC sets the appropriate electric field boundary conditions for the test item. On the surface of a perfectly conducting aircraft, the total tangential E-field vanishes so

$$E_{\text{scattered}}^{\text{tan}} = -E_{\text{incident}}^{\text{tan}}$$

The conditions are set in one part of the subroutine for any cell that defines a three-dimensional portion of the test item - E_x , E_y and then E_z . The setting of boundary conditions for the two dimensional surfaces is done in a separate part of the subroutine.

The values of the NOPE array for the cell and its surrounding cells are used to determine what the E-field value should be. If NOPE(I,J,K) is zero, no action is taken. For a cell with NOPE(I,J,K) = 4, the three surrounding cells are checked. If any one of the three is missing, the field is on the surface and E is set to $-E_{\text{incident}}^{\text{tan}}$. If all three cells are present E is not set, since it is then internal to the test item. Figure 6 indicates the situation for E_x on cell I,J,K. It is clear that if either cell ② ③ or ④ is not a part of the test item, E_x for cell ① would be on an external surface of the item. Similar situations can be defined for E_y and E_z .

In addition, if, for E_x , cell I,J,K is the outermost cell in either the J or the K directions, the E_x values at the far edges of this last cell are set equal to $-E_{\text{incident}}^{\text{tan}}$. The same is done for E_y , if the cell is the outermost in either the I or K directions and for E_z , in the I or J directions.

If NOPE defines a two-dimensional surface, the appropriate E-field components are set equal to $-E_{\text{incident}}^{\text{tan}}$. If the surface is in the X-Y plane, for example, E_x on the two sides of the rectangle defined by the cell and E_y on the other two sides are set equal to $-E_{\text{incident}}^{\text{tan}}$.

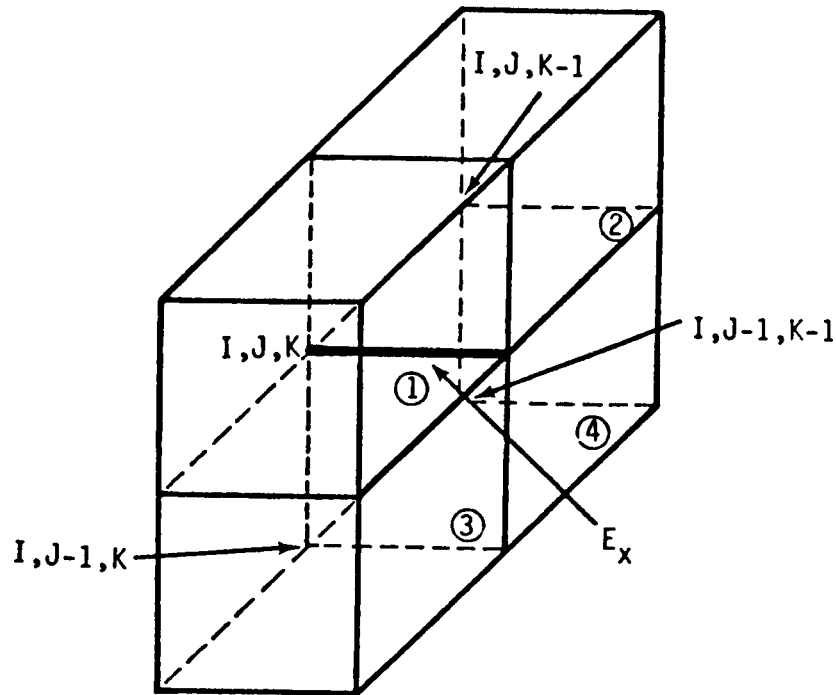


Figure 6. Determine E-Field Boundary Conditions

For a lossy dielectric aircraft or an aircraft composed, in part, by a lossy dielectric, the boundary condition is given by

$$\epsilon \frac{\partial \vec{E}^s}{\partial t} + \sigma \vec{E}^s = \vec{\nabla} \times \vec{H}^s - \sigma \vec{E}^i - (\epsilon - \epsilon_0) \frac{\partial \vec{E}^i}{\partial t}$$

This boundary condition holds within the lossy dielectric volume. It may be considered a volume boundary condition. In the limit, $\sigma \rightarrow \infty$, it becomes the surface boundary condition

$$E_{\text{scattered}}^{\text{tan}} = -E_{\text{incident}}^{\text{tan}}$$

for perfectly conducting surfaces.

Solid blocks (3-dimensional bodies) and plates (2-dimensional bodies) are treated much the same as for the perfectly conducting counterparts, except:

- 1) the volume boundary condition is used
- 2) the NOPE array values of 1,2,3,12,13,23,123, and 4 are replaced by 5,6,7,56,57,67,567 and 8 respectively
- 3) the plates have thickness TX, TY and TZ in the x, y and z directions respectively, where the thickness is expressed as a fraction of the cell dimension in the appropriate direction
- 4) the volume boundary condition is slightly modified, as discussed in Appendix A: Thin Plate Formalism, to more accurately model plates.

The generalized flow of the Subroutine EBC is given in Figure 7. The details of the flow through the perfectly conducting portion of EBC are given in Figure 8. The lossy dielectric portion follows the same approach. Subroutine EBC is called from EADV. EINCX, EINCY, and EINCZ are called by EBC.

Thin wires representing the lightning channels and interim wires (M=1 and 12) are also input in EBC.

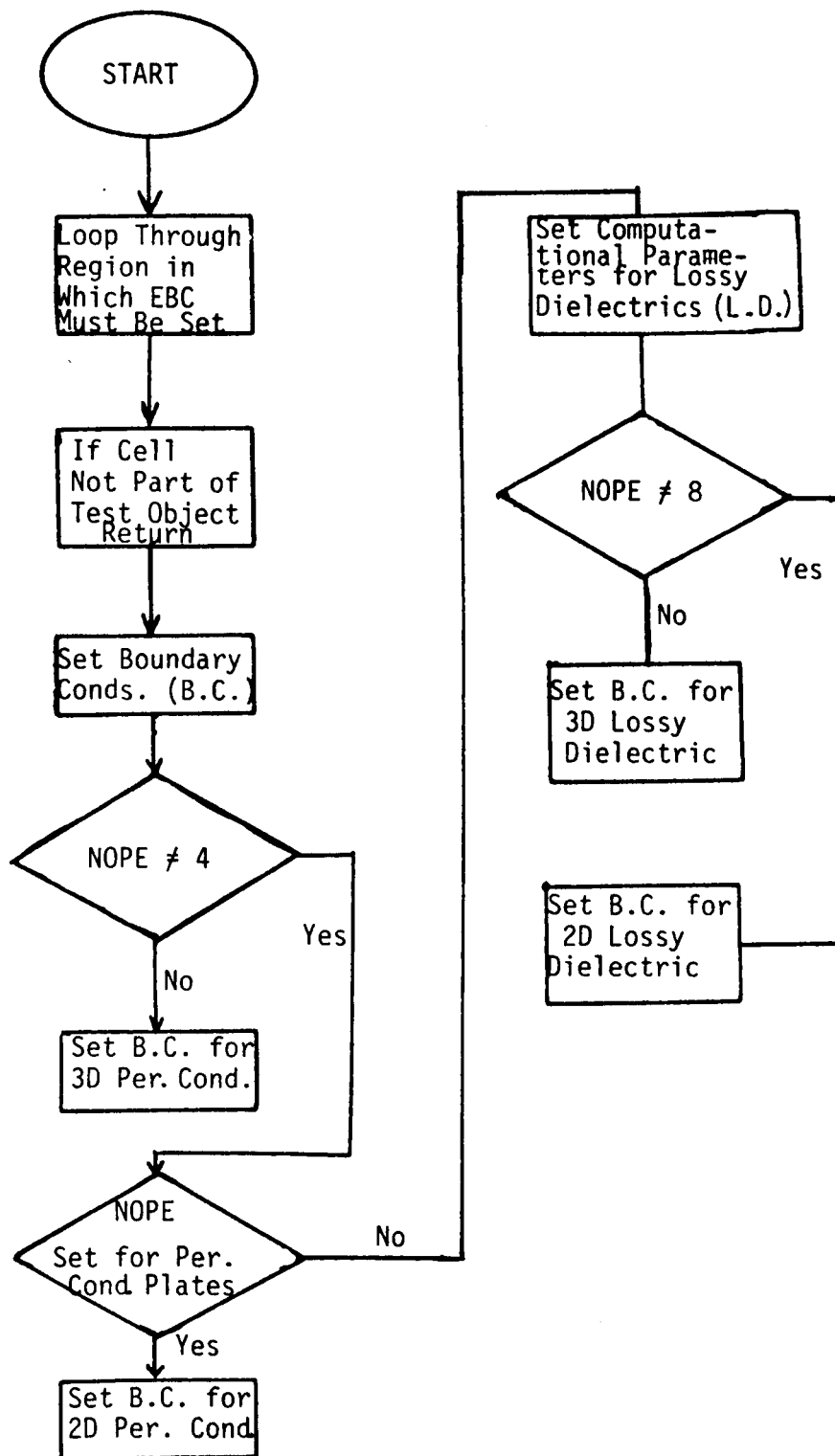


Figure 7. Generalized Flow Chart of Subroutine EBC

ORIGINAL PAGE IS
OF POOR QUALITY

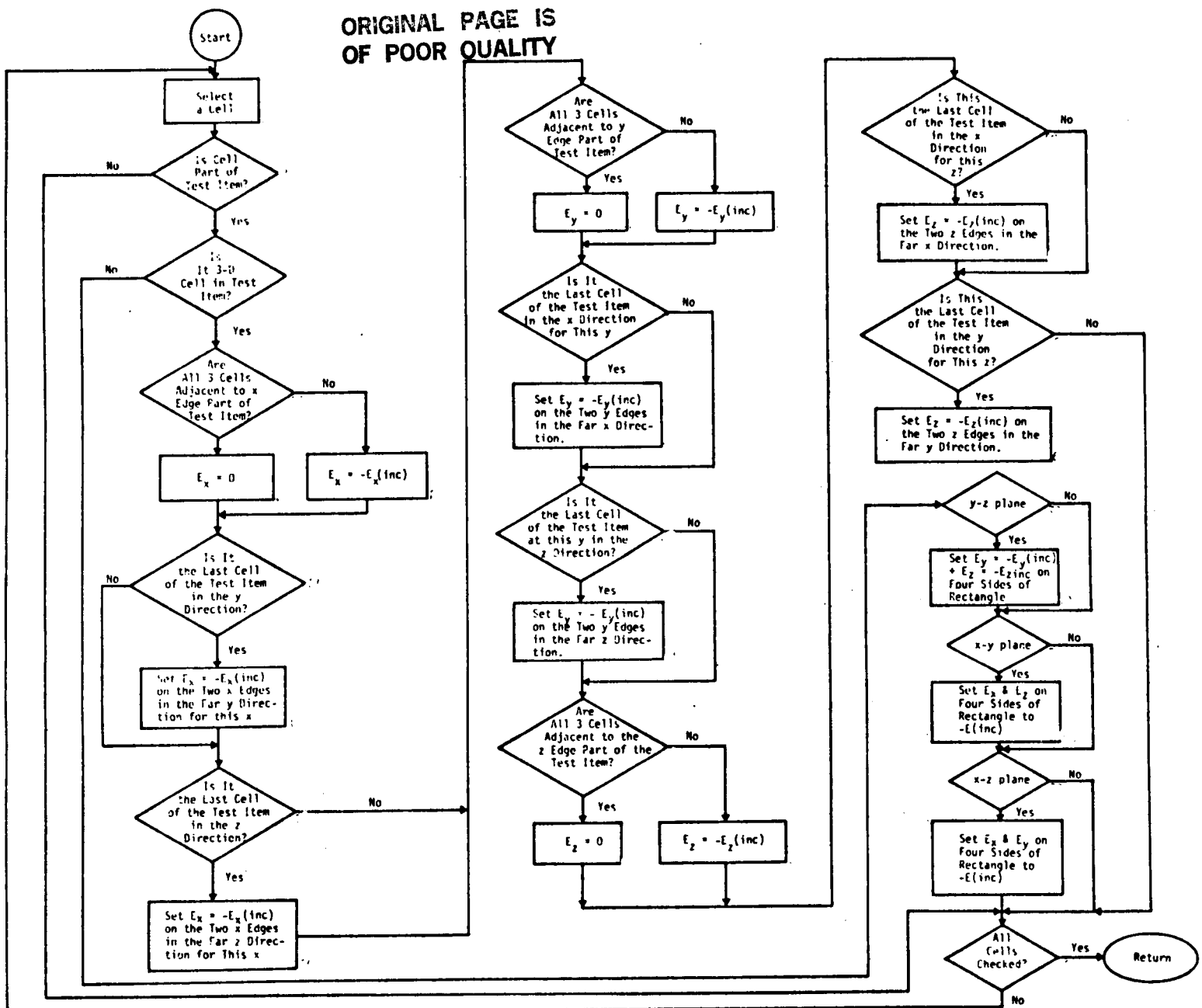


Figure 8. Flow Chart of Subroutine EBC

```

SUBROUTINE EBC
C
COMMON/EFIELD/EX(29,29,29),EY(29,29,29),EZ(29,29,29)
COMMON/HFIELD/HX(29,29,29),HY(29,29,29),HZ(29,29,29)
COMMON/GRID/X(28),Y(28),Z(28),X0(29),Y0(29),Z0(29),
1 DX(29),DY(29),DZ(29),DX0(28),DY0(28),DZ0(28),
2 DXI(29),DYI(29),DZI(29),DXOI(28),DYOI(28),DZOI(28)
COMMON/TSITEM/NOPE(29,29,29)
COMMON/EXTRAS/NX,NY,NZ,NX1,NY1,NZ1,N,M,MQ,DT,XMU,EPSO,EPS,NPTLIM,
1 NN,NPTS,LMAX,SIGMA,C,T,PI,EXPFAC,IP,TX,TY,TZ,AMP,ALPHA,BETA,IDLS
C
IF(IDLS.EQ.0) GO TO 1
IF(M.NE.1) GO TO 2
C
C UNEXPANDED LIGHTNING CHANNELS
C
DO 3 J=13,28
EY(8,J,15)=0.0
3 CONTINUE
C
DO 4 J=1,11
EY(22,J,15)=0.0
4 CONTINUE
C
GO TO 1
C
2 CONTINUE
IF(M.NE.2) GO TO 5
C
C EXPANDED LIGHTNING CHANNELS
C
DO 6 J=9,28
EY(5,J,13)=0.0
6 CONTINUE
C
GO TO 1
C
5 CONTINUE
C
C
C
1 CONTINUE
C
C
C
C LOOP OVER REGION IN WHICH EBC MUST BE SET
C
DO 500 I=1,NX1
DO 500 J=1,NY1
DO 500 K=1,NZ1
C
IF(M.EQ.1) GO TO 12
TX=0.008
TY=0.02
TZ=0.02

```

```

12 CONTINUE
C
    IF(NOPE(I,J,K).EQ.0)GO TO 500
C
    IF (NOPE(I,J,K).NE.4) GO TO 60
C
    SET BOUNDARY CONDITIONS ON PERFECTLY CONDUCTING 3-D BODY
C
    SET EX
C
    IF((J-1).EQ.0.OR.(K-1).EQ.0) GO TO 18
    IF(NOPE(I,J,K-1).EQ.4.AND.NOPE(I,J-1,K-1).EQ.4.AND.
1 NOPE(I,J-1,K).EQ.4)GO TO 10
    EX(I,J,K)=-EINCX(I,J,K)
10 CONTINUE
    IF(NOPE(I,J+1,K).EQ.4)GO TO 15
    EX(I,J+1,K+1)=-EINCX(I,J+1,K+1)
    EX(I,J+1,K)=-EINCX(I,J+1,K)
15 IF(NOPE(I,J,K+1).EQ.4)GO TO 20
    EX(I,J,K+1)=-EINCX(I,J,K+1)
    EX(I,J+1,K+1)=-EINCX(I,J+1,K+1)
C
    SET EY
C
18 CONTINUE
    IF((I-1).EQ.0.OR.(K-1).EQ.0) GO TO 38
20 IF(NOPE(I-1,J,K).EQ.4.AND.NOPE(I-1,J,K-1).EQ.4.AND.
1 NOPE(I,J,K-1).EQ.4)GO TO 30
    EY(I,J,K)=-EINCY(I,J,K)
30 CONTINUE
    IF(NOPE(I+1,J,K).EQ.4)GO TO 35
    EY(I+1,J,K+1)=-EINCY(I+1,J,K+1)
    EY(I+1,J,K)=-EINCY(I+1,J,K)
35 IF(NOPE(I,J,K+1).EQ.4)GO TO 40
    EY(I,J,K+1)=-EINCY(I,J,K+1)
    EY(I+1,J,K+1)=-EINCY(I+1,J,K+1)
C
    SET EZ
C
38 CONTINUE
    IF((I-1).EQ.0.OR.(J-1).EQ.0) GO TO 500
40 IF(NOPE(I-1,J,K).EQ.4.AND.NOPE(I-1,J-1,K).EQ.4.AND.
1NOPE(I,J-1,K).EQ.4)GO TO 50
    EZ(I,J,K)=-EINCZ(I,J,K)
50 CONTINUE
    IF(NOPE(I+1,J,K).EQ.4)GO TO 55
    EZ(I+1,J,K)=-EINCZ(I+1,J,K)
    EZ(I+1,J+1,K)=-EINCZ(I+1,J+1,K)
55 IF(NOPE(I,J+1,K).EQ.4)GO TO 500
    EZ(I+1,J+1,K)=-EINCZ(I+1,J+1,K)
    EZ(I,J+1,K)=-EINCZ(I,J+1,K)
    GO TO 500
C
60 CONTINUE
C

```

```

      IF(NOPE(I,J,K).NE.1.AND.NOPE(I,J,K).NE.12.AND.NOPE(I,J,K).NE.13.
1    AND.NOPE(I,J,K).NE.123.AND.NOPE(I,J,K).NE.2.AND.
2    NOPE(I,J,K).NE.23.AND.NOPE(I,J,K).NE.3)GO TO 510
C
C    SET BOUNDARY CONDITIONS ON PERFECTLY CONDUCTING 2-D BODY
C
      IF(NOPE(I,J,K).NE.1.AND.NOPE(I,J,K).NE.12.AND.NOPE(I,J,K).NE.13.
1    AND.NOPE(I,J,K).NE.123) GO TO 70
C
C    Y-Z PLANE
C
      EY(I,J,K+1)=-EINCY(I,J,K+1)
      EY(I,J,K)=-EINCY(I,J,K)
      EZ(I,J+1,K)=-EINCZ(I,J+1,K)
      EZ(I,J,K)=-EINCZ(I,J,K)
C
      IF(NOPE(I,J,K).EQ.12.OR.NOPE(I,J,K).EQ.123) GO TO 70
      IF(NOPE(I,J,K).EQ.13) GO TO 80
      GO TO 500
C
70  IF(NOPE(I,J,K).NE.2.AND.NOPE(I,J,K).NE.23.AND.
1    NOPE(I,J,K).NE.12.AND.NOPE(I,J,K).NE.123) GO TO 80
C
C    X-Z PLANE
C
      EX(I,J,K+1)=-EINCX(I,J,K+1)
      EX(I,J,K)=-EINCX(I,J,K)
      EZ(I+1,J,K)=-EINCZ(I+1,J,K)
      EZ(I,J,K)=-EINCZ(I,J,K)
C
      IF(NOPE(I,J,K).EQ.23.OR.NOPE(I,J,K).EQ.123) GO TO 80
      GO TO 500
C
80  CONTINUE
C
C    X-Y PLANE
C
      EX(I,J+1,K)=-EINCX(I,J+1,K)
      EX(I,J,K)=-EINCX(I,J,K)
      EY(I+1,J,K)=-EINCY(I+1,J,K)
      EY(I,J,K)=-EINCY(I,J,K)
C
      GO TO 500
C
510 CONTINUE
C
C    SET COMPUTATIONAL PARAMETERS
C
8    RSIGMA=1./SIGMA
      EPEFFX=(1.0-TX)*EPSO+(TX*EPS)
      EPEFFY=(1.0-TY)*EPSO+(TY*EPS)
      EPEFFZ=(1.0-TZ)*EPSO+(TZ*EPS)
      EXPON=EXP((-SIGMA*DT)/EPS)
      EXPDX=EXP((-SIGMA*TX*DT)/EPEFFX)
      EXPDY=EXP((-SIGMA*TY*DT)/EPEFFY)

```

```

EXPDZ=EXP((-SIGMA*TZ*DT)/EPEFFZ)
XEXP=1.-EXPON
XEXPX=1.-EXPDX
XEXPY=1.-EXPDY
XEXPZ=1.-EXPDZ
C
IF(NOPE(I,J,K).NE.8)GO TO 600
C
SET BOUNDARY CONDITIONS ON LOSSY DIELECTRIC 3-D BODY
C
SET EX
C
IF((J-1).EQ.0.OR.(K-1).EQ.0) GO TO 200
EX(I,J,K)=(EX(I,J,K)*EXPON)+((XEXP)*(-EINCX(I,J,K)
1      -((EPS-EPSO)*RSIGMA)*DEINCX(I,J,K)
2      +(((HZ(I,J,K)-HZ(I,J-1,K))*DYI(J))
3      -((HY(I,J,K)-HY(I,J,K-1))*DZI(K)))*RSIGMA))
C
IF(NOPE(I,J+1,K).EQ.8)GO TO 150
C
EX(I,J+1,K)=(EX(I,J+1,K)*EXPON)+((XEXP)*(-EINCX(I,J+1,K)
1      -((EPS-EPSO)*RSIGMA)*DEINCX(I,J+1,K)
2      +(((HZ(I,J+1,K)-HZ(I,J,K))*DYI(J+1))
3      -((HY(I,J+1,K)-HY(I,J+1,K-1))*DZI(K)))*RSIGMA))
C
EX(I,J+1,K+1)=(EX(I,J+1,K+1)*EXPON)+((XEXP)*(-EINCX(I,J+1,K+1)
1      -((EPS-EPSO)*RSIGMA)*DEINCX(I,J+1,K+1)
2      +(((HZ(I,J+1,K+1)-HZ(I,J,K+1))*DYI(J+1))
3      -((HY(I,J+1,K+1)-HY(I,J+1,K))*DZI(K+1)))*RSIGMA))
C
150 IF(NOPE(I,J,K+1).EQ.8)GO TO 200
C
EX(I,J,K+1)=(EX(I,J,K+1)*EXPON)+((XEXP)*(-EINCX(I,J,K+1)
1      -((EPS-EPSO)*RSIGMA)*DEINCX(I,J,K+1)
2      +(((HZ(I,J,K+1)-HZ(I,J-1,K+1))*DYI(J))
3      -((HY(I,J,K+1)-HY(I,J,K))*DZI(K+1)))*RSIGMA))
C
EX(I,J+1,K+1)=(EX(I,J+1,K+1)*EXPON)+((XEXP)*(-EINCX(I,J+1,K+1)
1      -((EPS-EPSO)*RSIGMA)*DEINCX(I,J+1,K+1)
2      +(((HZ(I,J+1,K+1)-HZ(I,J,K+1))*DYI(J+1))
3      -((HY(I,J+1,K+1)-HY(I,J+1,K))*DZI(K+1)))*RSIGMA))
C
SET EY
C
200 CONTINUE
C
IF((I-1).EQ.0.OR.(K-1).EQ.0) GO TO 300
EY(I,J,K)=(EY(I,J,K)*EXPON)+((XEXP)*(-EINCY(I,J,K)
1      -((EPS-EPSO)*RSIGMA)*DEINCY(I,J,K)
2      +(((HX(I,J,K)-HX(I,J,K-1))*DZI(K))
3      -((HZ(I,J,K)-HZ(I-1,J,K))*DXI(I)))*RSIGMA))
C
IF(NOPE(I+1,J,K).EQ.8)GO TO 250
C
EY(I+1,J,K+1)=(EY(I+1,J,K+1)*EXPON)+((XEXP)*(-EINCY(I+1,J,K+1)

```



```

1 -((EPS-EP SO)*RSIGMA)*DEINCY(I+1,J,K+1)
2 +(((HX(I+1,J,K+1)-HX(I+1,J,K))*DZI(K+1))
3 -((HZ(I+1,J,K+1)-HZ(I,J,K+1))*DXI(I+1)))*RSIGMA))
C
EY(I+1,J,K)=(EY(I+1,J,K)*EXPON)+((XEXP)*(-EINCY(I+1,J,K)
1 -((EPS-EP SO)*RSIGMA)*DEINCY(I+1,J,K)
2 +(((HX(I+1,J,K)-HX(I+1,J,K-1))*DZI(K))
3 -((HZ(I+1,J,K)-HZ(I,J,K))*DXI(I+1)))*RSIGMA))
C
250 IF(NOPE(I,J,K+1).EQ.8)GO TO 300
C
EY(I,J,K+1)=(EY(I,J,K+1)*EXPON)+((XEXP)*(-EINCY(I,J,K+1)
1 -((EPS-EP SO)*RSIGMA)*DEINCY(I,J,K+1)
2 +(((HX(I,J,K+1)-HX(I,J,K))*DZI(K+1))
3 -((HZ(I,J,K+1)-HZ(I-1,J,K+1))*DXI(I)))*RSIGMA))
C
EY(I+1,J,K+1)=(EY(I+1,J,K+1)*EXPON)+((XEXP)*(-EINCY(I+1,J,K+1)
1 -((EPS-EP SO)*RSIGMA)*DEINCY(I+1,J,K+1)
2 +(((HX(I+1,J,K+1)-HX(I+1,J,K))*DZI(K+1))
3 -((HZ(I+1,J,K+1)-HZ(I,J,K+1))*DXI(I+1)))*RSIGMA))
C
C SET EZ
C
300 CONTINUE
C
IF((I-1).EQ.0.OR.(J-1).EQ.0) GO TO 500
EZ(I,J,K)=(EZ(I,J,K)*EXPON)+((XEXP)*(-EINCZ(I,J,K)
1 -((EPS-EP SO)*RSIGMA)*DEINCZ(I,J,K)
2 +(((HY(I,J,K)-HY(I-1,J,K))*DXI(I))
3 -((HX(I,J,K)-HX(I,J-1,K))*DYI(J)))*RSIGMA))
C
IF(NOPE(I+1,J,K).EQ.8)GO TO 350
C
EZ(I+1,J,K)=(EZ(I+1,J,K)*EXPON)+((XEXP)*(-EINCZ(I+1,J,K)
1 -((EPS-EP SO)*RSIGMA)*DEINCZ(I+1,J,K)
2 +(((HY(I+1,J,K)-HY(I,J,K))*DXI(I+1))
3 -((HX(I+1,J,K)-HX(I+1,J-1,K))*DYI(J)))*RSIGMA))
C
EZ(I+1,J+1,K)=(EZ(I+1,J+1,K)*EXPON)+((XEXP)*(-EINCZ(I+1,J+1,K)
1 -((EPS-EP SO)*RSIGMA)*DEINCZ(I+1,J+1,K)
2 +(((HY(I+1,J+1,K)-HY(I,J+1,K))*DXI(I+1))
3 -((HX(I+1,J+1,K)-HX(I+1,J,K))*DYI(J+1)))*RSIGMA))
C
350 IF(NOPE(I,J+1,K).EQ.8)GO TO 500
C
EZ(I+1,J+1,K)=(EZ(I+1,J+1,K)*EXPON)+((XEXP)*(-EINCZ(I+1,J+1,K)
1 -((EPS-EP SO)*RSIGMA)*DEINCZ(I+1,J+1,K)
2 +(((HY(I+1,J+1,K)-HY(I,J+1,K))*DXI(I+1))
3 -((HX(I+1,J+1,K)-HX(I+1,J,K))*DYI(J+1)))*RSIGMA))
C
EZ(I,J+1,K)=(EZ(I,J+1,K)*EXPON)+((XEXP)*(-EINCZ(I,J+1,K)
1 -((EPS-EP SO)*RSIGMA)*DEINCZ(I,J+1,K)
2 +(((HY(I,J+1,K)-HY(I-1,J+1,K))*DXI(I))
3 -((HX(I,J+1,K)-HX(I,J,K))*DYI(J+1)))*RSIGMA))
C

```

```

      GO TO 500
C
600 CONTINUE
C
C      SET BOUNDARY CONDITIONS ON LOSSY DIELECTRIC 2-D BODY
C
      IF((I-1).EQ.0.OR.(J-1).EQ.0.OR.(K-1).EQ.0) GO TO 500
      IF(NOPE(I,J,K).NE.5.AND.NOPE(I,J,K).NE.56.AND.NOPE(I,J,K).
1 NE.57.AND.NOPE(I,J,K).NE.567) GO TO 700
C
C      Y-Z PLANE
C
      EY(I,J,K+1)=(EY(I,J,K+1)*EXPDX)+((XEXPX)*(-EINCY(I,J,K+1)
1 -((EPS-EPSO)*RSIGMA)*DEINCY(I,J,K+1)
2 +(((HX(I,J,K+1)-HX(I,J,K))*DZI(K+1))
3 -((HZ(I,J,K+1)-HZ(I-1,J,K+1))*DXI(I))*RSIGMA*TX))
C
      EY(I,J,K)=(EY(I,J,K)*EXPDX)+((XEXPX)*(-EINCY(I,J,K)
1 -((EPS-EPSO)*RSIGMA)*DEINCY(I,J,K)
2 +(((HX(I,J,K)-HX(I,J,K-1))*DZI(K))
3 -((HZ(I,J,K)-HZ(I-1,J,K))*DXI(I))*RSIGMA*TX))
C
      EZ(I,J+1,K)=(EZ(I,J+1,K)*EXPDX)+((XEXPX)*(-EINCZ(I,J+1,K)
1 -((EPS-EPSO)*RSIGMA)*DEINCZ(I,J+1,K)
2 +(((HY(I,J+1,K)-HY(I-1,J+1,K))*DXI(I))
3 -((HX(I,J+1,K)-HX(I,J,K))*DYI(J+1))*RSIGMA*TX))
C
      EZ(I,J,K)=(EZ(I,J,K)*EXPDX)+((XEXPX)*(-EINCZ(I,J,K)
1 -((EPS-EPSO)*RSIGMA)*DEINCZ(I,J,K)
2 +(((HY(I,J,K)-HY(I-1,J,K))*DXI(I))
3 -((HX(I,J,K)-HX(I,J-1,K))*DYI(J))*RSIGMA*TX))
C
      IF(NOPE(I,J,K).EQ.56) GO TO 700
      IF(NOPE(I,J,K).EQ.57) GO TO 800
      IF(NOPE(I,J,K).EQ.567) GO TO 700
      GO TO 500
C
700 IF(NOPE(I,J,K).NE.6.AND.NOPE(I,J,K).NE.67.AND.
1 NOPE(I,J,K).NE.56.AND.NOPE(I,J,K).NE.567) GO TO 800
C
C      X-Z PLANE
C
      EX(I,J,K)=(EX(I,J,K)*EXPDY)+((XEXPY)*(-EINCX(I,J,K)
1 -((EPS-EPSO)*RSIGMA)*DEINCX(I,J,K)
2 +(((HZ(I,J,K)-HZ(I,J-1,K))*DYI(J))
3 -((HY(I,J,K)-HY(I,J,K-1))*DZI(K))*RSIGMA*TY))
C
      EX(I,J,K+1)=(EX(I,J,K+1)*EXPDY)+((XEXPY)*(-EINCX(I,J,K+1)
1 -((EPS-EPSO)*RSIGMA)*DEINCX(I,J,K+1)
2 +(((HZ(I,J,K+1)-HZ(I,J-1,K+1))*DYI(J))
3 -((HY(I,J,K+1)-HY(I,J,K))*DZI(K+1))*RSIGMA*TY))
C
      EZ(I+1,J,K)=(EZ(I+1,J,K)*EXPDY)+((XEXPY)*(-EINCZ(I+1,J,K)
1 -((EPS-EPSO)*RSIGMA)*DEINCZ(I+1,J,K)
2 +(((HY(I+1,J,K)-HY(I,J,K))*DXI(I+1))

```

```

C      3      -((HX(I+1,J,K)-HX(I+1,J-1,K))*DYI(J))*RSIGMA*TY))
      EZ(I,J,K)=(EZ(I,J,K)*EXPDY)+(XEXPY)*(-EINCZ(I,J,K)
1      -((EPS-EPSO)*RSIGMA)*DEINCZ(I,J,K)
2      +(((HY(I,J,K)-HY(I-1,J,K))*DXI(I))
3      -((HX(I,J,K)-HX(I,J-1,K))*DYI(J))*RSIGMA*TY))
C
      IF(NOPE(I,J,K).EQ.67.OR.NOPE(I,J,K).EQ.567) GO TO 800
      GO TO 500
C
800 CONTINUE
C
C      X-Y PLANE
C
      EX(I,J,K)=(EX(I,J,K)*EXPDZ)+(XEXPZ)*(-EINCX(I,J,K)
1      -((EPS-EPSO)*RSIGMA)*DEINCX(I,J,K)
2      +(((HZ(I,J,K)-HZ(I,J-1,K))*DYI(J))
3      -((HY(I,J,K)-HY(I,J,K-1))*DZI(K))*RSIGMA*TZ))
C
      EX(I,J+1,K)=(EX(I,J+1,K)*EXPDZ)+(XEXPZ)*(-EINCX(I,J+1,K)
1      -((EPS-EPSO)*RSIGMA)*DEINCX(I,J+1,K)
2      +(((HZ(I,J+1,K)-HZ(I,J,K))*DYI(J+1))
C      3      -((HY(I,J+1,K)-HY(I,J+1,K-1))*DZI(K))*RSIGMA*TZ))
      EY(I,J,K)=(EY(I,J,K)*EXPDZ)+(XEXPZ)*(-EINCY(I,J,K)
1      -((EPS-EPSO)*RSIGMA)*DEINCY(I,J,K)
2      +(((HX(I,J,K)-HX(I,J,K-1))*DZI(K))
3      -((HZ(I,J,K)-HZ(I-1,J,K))*DXI(I))*RSIGMA*TZ))
C
      EY(I+1,J,K)=(EY(I+1,J,K)*EXPDZ)+(XEXPZ)*(-EINCY(I+1,J,K)
1      -((EPS-EPSO)*RSIGMA)*DEINCY(I+1,J,K)
2      +(((HX(I+1,J,K)-HX(I+1,J,K-1))*DZI(K))
3      -((HZ(I+1,J,K)-HZ(I,J,K))*DXI(I+1))*RSIGMA*TZ))
C
500 CONTINUE
501 CONTINUE
C
      NWIRE=1
      IF(NWIRE.EQ.0) GO TO 550
      IF(M.NE.2) GO TO 550
C
C      B1 TO B2
C
      I=4
      K=13
      DO 900 J=10,14
      EY(I,J,K)=-EINCY(I,J,K)
900 CONTINUE
C
      EX(4,10,13)=-EINCX(4,10,13)
      EZ(5,10,13)=-EINCZ(5,10,13)
      EZ(5,10,14)=-EINCZ(5,10,14)
      EX(5,10,15)=-EINCX(5,10,15)
      EZ(6,10,15)=-EINCZ(6,10,15)
      EZ(6,10,16)=-EINCZ(6,10,16)

```

```

C
C   B2 THROUGH B4
C
C   EX(6,10,17)=-EINCX(6,10,17)
C   EZ(7,10,17)=-EINCZ(7,10,17)
C
C       J=10
C       K=18
C   DO 901 I=7,10
C       EX(I,J,K)=-EINCX(I,J,K)
901 CONTINUE
C
C   AFT OF B4 THROUGH B6
C
C   EZ(11,10,18)=-EINCZ(11,10,18)
C
C       J=10
C       K=19
C   DO 902 I=11,16
C       EX(I,J,K)=-EINCX(I,J,K)
902 CONTINUE
C
C   AFT OF B6 TO B8
C
C       I=17
C       K=19
C   DO 903 J=10,13
C       EY(I,J,K)=-EINCY(I,J,K)
903 CONTINUE
C
C   EX(17,14,19)=-EINCX(17,14,19)
C   EY(18,13,19)=-EINCY(18,13,19)
C   EX(18,13,19)=-EINCX(18,13,19)
C   LY(19,13,19)=-EINCY(19,13,19)
C   EX(19,14,19)=-EINCX(19,14,19)
C
C   B8 ON TO END
C
C   EX(20,14,19)=-EINCX(20,14,19)
C   EZ(21,14,18)=-EINCZ(21,14,18)
C   EZ(21,14,17)=-EINCZ(21,14,17)
C
C       J=14
C       K=17
C   DO 904 I=21,25
C       EX(I,J,K)=-EINCX(I,J,K)
904 CONTINUE
C
C   EY(26,13,17)=-EINCY(26,13,17)
C   EY(26,12,17)=-EINCY(26,12,17)
C   EY(26,11,17)=-EINCY(26,11,17)
C
C   EX(26,11,17)=-EINCX(26,11,17)
C
C   EY(27,11,17)=-EINCY(27,11,17)

```

```
      EY(27,12,17)=-EINCY(27,12,17)
C
      EX(27,13,17)=-EINCX(27,13,17)
C
550  CONTINUE
C
      RETURN
C
      END
```

3.5.5 Subroutine ERAD

Subroutine ERAD evaluates the expression

$$E(R, D, EL1, EL2, EL3, L1, L2, L3, TH) = R/D * \quad (1)$$

$$\begin{aligned} & (EL1 * (L2 - L3) * (L2 * L3 + TH * (L2 + L3) + TH**2) + \\ & EL2 * (L3 - L1) * (L3 * L1 + TH * (L3 + L1) + TH**2) + \\ & EL3 * (L1 - L2) * (L1 * L2 + TH * (L1 + L2) + TH**2)). \end{aligned}$$

It substitutes the results for the EADV-determined values of the field components which are on the surface of the problem cell space (shown in Figure 9) according to the prescription:

$$EX(I, 1, K) = E(RXY(I, K, 1), D, EX(I, 2, K), EXYD(I, K, 2),$$

$$EXYD(I, K, 3), L1, L2, L3, THXY(I, K, 1)) \quad (2a)$$

$$\begin{aligned} & \cdot \\ & \cdot \\ & \cdot \end{aligned} \quad \begin{aligned} & \cdot \\ & \cdot \\ & \cdot \end{aligned}$$

$$EZ(I, NY, K) = E(RZY(I, K, 2), D, EZ(I, NY1, K), EZYU(I, K, 2),$$

$$EZYU(I, K, 3), L1, L2, L3, THZY(I, K, 2)) \quad (2b)$$

This substitution imposes a far field behavior on the outermost region of the problem cell space. Equation (1) is derived directly from the assumption that the outermost electric field components of Figure 9 and the corresponding electric fields that are one cell in from the outer surface, behave as far fields,* that is,

$$E = f(\theta, \phi)g(t-r/c)/r \text{ or equivalently} \quad (3)$$

$$E_r = f(\theta, \phi)g(t-r/c)$$

*As a result, fields on the outermost edges are not written over, but this does not effect the interior results.

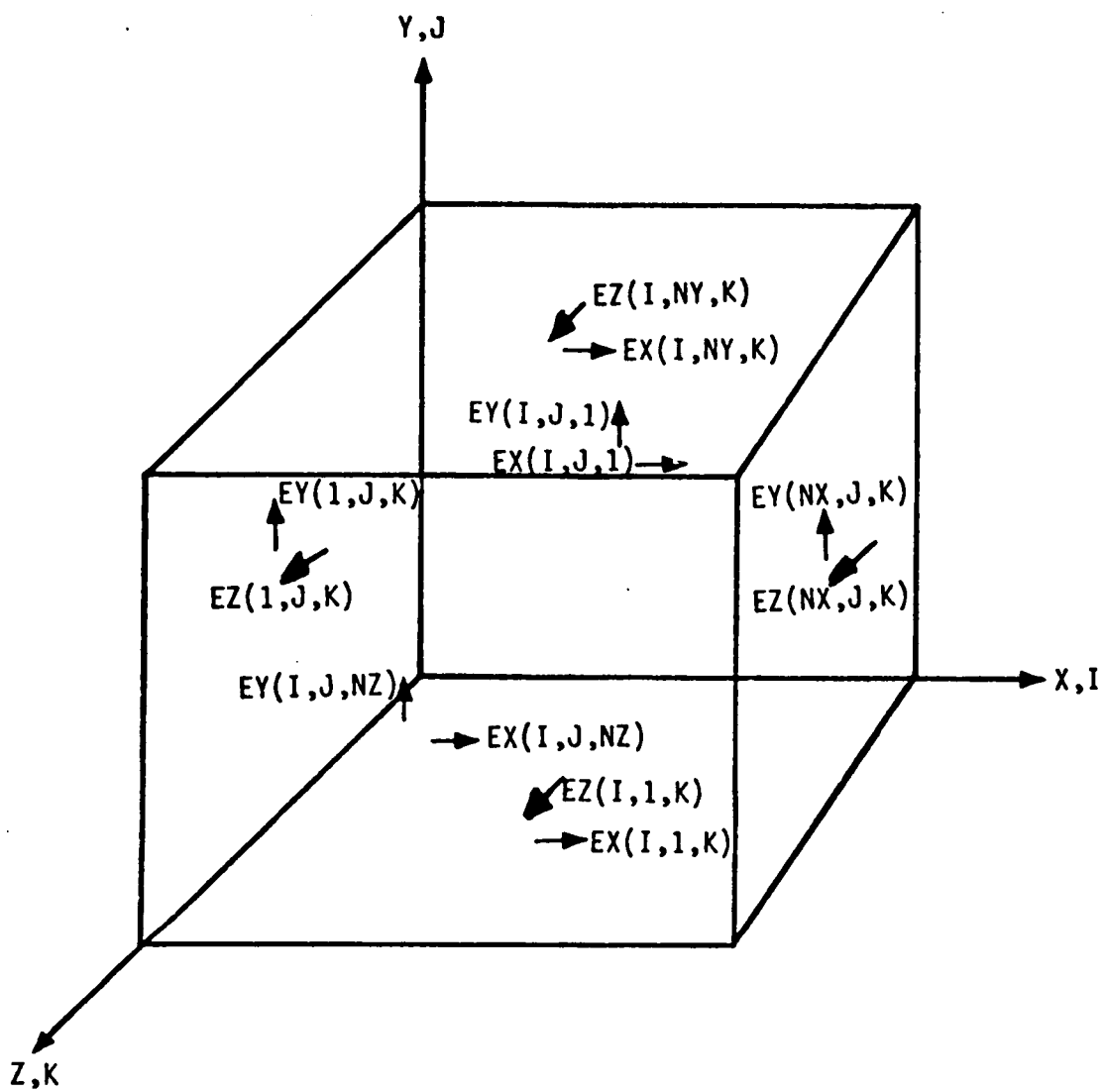


Figure 9. Field Components at the Outer Boundary of the Cell Space

Thus, taking $E_y^{n+1/2}(I,J,K)$ as an example,

$$R_{yx}(1,J,K) E_y^{n+1/2}(1,J,K) \equiv e_y(J,K,[(n+1/2)\Delta t - R_{yx}(1,J,K)/c]) \quad (4)$$

$$R_{yx}(2,J,K) E_y^{n+1/2}(2,J,K) \equiv e_y(J,K,[(n+1/2)\Delta t - R_{yx}(2,J,K)/c]) \quad (5)$$

where R_{yx} is the distance to the E_y component.

What is desired is $E_y^{n+1/2}(I,J,K)$ expressed in terms of $E_y^{n+1/2-\ell_i}(2,J,K)$, i.e., the outer boundary field component expressed in terms of an interior corresponding field component at various time offsets(ℓ_i). Equation (1) is, therefore, rewritten as,

$$\begin{aligned} R_{yx}(1,J,K) E_y^{n+1/2}(1,J,K) &= e_y \left(J,K, \left[(n-1/2)\Delta t + \Delta t \right. \right. \\ &\quad \left. \left. - \frac{R_{yx}(2,J,K)}{c} - \frac{R_{yx}(1,J,K) - R_{yz}(2,J,K)}{c} \right] \right) \\ &= e_y \left(J,K, \left[(n-1/2)\Delta t - \frac{R_{yx}(2,J,K)}{c} + \theta_{yx}(J,K)\Delta t \right] \right) \end{aligned} \quad (6)$$

where

$$\theta_{yx} \equiv 1 - \frac{R_{yx}(1,J,K) - R_{yx}(2,J,K)}{c\Delta t} \quad (7)$$

and Equation (2) is rewritten with a time shift ℓ_i as,

$$\begin{aligned} R(2,J,K) E_y^{n-1/2-\ell_i}(2,J,K) &= \\ e_y \left(J,K, \left[(n-1/2)\Delta t - \frac{R_{yx}(2,J,K)}{c} - \ell_i\Delta t \right] \right) \end{aligned} \quad (8)$$

It can be seen that Equations (6) and (8) are identical except for ℓ_i in place of θ_{yx} in Equation (8). Either Equation (6) or (8) can be expanded in terms of the time $\ell_i \Delta t$ or $\theta_{yx} \Delta t$. The first three terms of Maclaurin's series are used so that

$$R_{yx}(1,J,K) E_y^{n+1/2}(1,J,K) = A + B (\theta_{yx} \Delta t) + C (\theta_{yx} \Delta t)^2 \quad (9)$$

$$R_{yx}(2,J,K) E_y^{n-1/2-\ell_i}(2,J,K) = A + B (\ell_i \Delta t) + C (\ell_i \Delta t)^2. \quad (10)$$

If three different values of ℓ_i are selected, then, from Equation (10), we obtain three equations for the three unknown A, B and C in terms of the three values of $E_y^{n-1/2-\ell_i}$, the three values of ℓ_i and $R_{yx}(2,J,K)$, all of which are known if $\ell_i \geq -1$. Therefore, A, B and C can be determined and are

$$A = [E_y^{n-1/2-\ell_i}(2,J,K) \ell_2 \ell_3 (\ell_2 - \ell_3) \quad (11)$$

$$+ E_y^{n-1/2-\ell_i}(2,J,K) \ell_3 \ell_1 (\ell_3 - \ell_1)$$

$$+ E_y^{n-1/2-\ell_i}(2,J,K) \ell_1 \ell_2 (\ell_1 - \ell_2)] R_{yx}(2,J,K)/D$$

$$B = [E_y^{n-1/2-\ell_1}(2,J,K) (\ell_2^2 - \ell_3^2) \quad (12)$$

$$+ E_y^{n-1/2-\ell_3}(2,J,K) (\ell_1^2 - \ell_2^2)] R_{yx}(2,J,K)/D \Delta t$$

$$C = [E_y^{n-1/2-\ell_1}(2,J,K) (\ell_2 - \ell_3) \quad (13)$$

$$+ E_y^{n-1/2-\ell_2}(2,J,K) (\ell_3 - \ell_1)$$

$$+ E_y^{n-1/2-\ell_3}(2,J,K) (\ell_1 - \ell_2)] R_{yx}(2,J,K)/D \Delta t^2$$

where D is the determinant

$$D = l_2 l_3 (l_2 - l_3) + l_3 l_1 (l_3 - l_1) + l_1 l_2 (l_1 - l_2) \quad (14)$$

Knowing A, B and C, Equation (9) immediately yields

$$E_y^{n+1/2}(1,J,K) = \frac{A + B(\theta_{yx} \Delta t) + C(\theta_{yx} \Delta t)^2}{R_{yx}(1,J,K)} \quad (15)$$

This is the desired result of $E_y^{n+1/2}(I,J,K)$ expressed in terms of $E_y^{n-1/2-l_i}$ and other known quantities, namely $R_{yx}(I,J,K)$, $R_{yx}(2,J,K)$ and l_1 , l_2 and l_3 .

The last equation, Equation (15), is identical to the expression, Equation (1), that ERAD evaluates.

For example, the terms $EX(I,2,K)$, $EXYD(I,K,2)$, and $EXYD(I,K,3)$ in Equation (2a) are the backstored values of EX at $J = 2$ that are used to calculate $EX(I,1,K)$. They are found in Subroutine EADV. The quantities D, L1, L2, L3, $THXY(I,K,1)$ correspond to D, l_1 , l_2 , and $\theta_{xy}(I,K)$. D is calculated in ERAD as are l_1 , l_2 and l_3 and the corresponding stored values of E one cell in from the outer boundary. The l_i 's are spread enough in time to correspond to the propagation time across the expanded cell boundary. Reference 2 details what requirements on the l_i and, hence, on N2 and N3 in EADV must be met. Only when constant mesh size is used can the $l_1 = -1$, $l_2 = 0$, $l_3 = 1$. In this case, the expression in Equation (1) goes over exactly to the expression in Section 2.3. The value of $R_{xy}(I,K,1)$ (which represents $R_{xy}(I,K,2)/R_{xy}(I,K,1)$ in this subroutine) is determined in Subroutine SETUP, as is the value of $THXY(I,K,1)$. Since they are geometry dependent quantities, they need only be found once, hence, their presence in SETUP.

Flow for Subroutine ERAD is as shown in Figure 10. Local variables are described in Table 5. Subroutine ERAD is called by EADV.

Table 5. Local Variables - Subroutine ERAD

Variable	
<u>NAME</u>	<u>Description</u>
D	Determinant
L1	Time shifts for backstored data in number of cycles
L2	
L3	

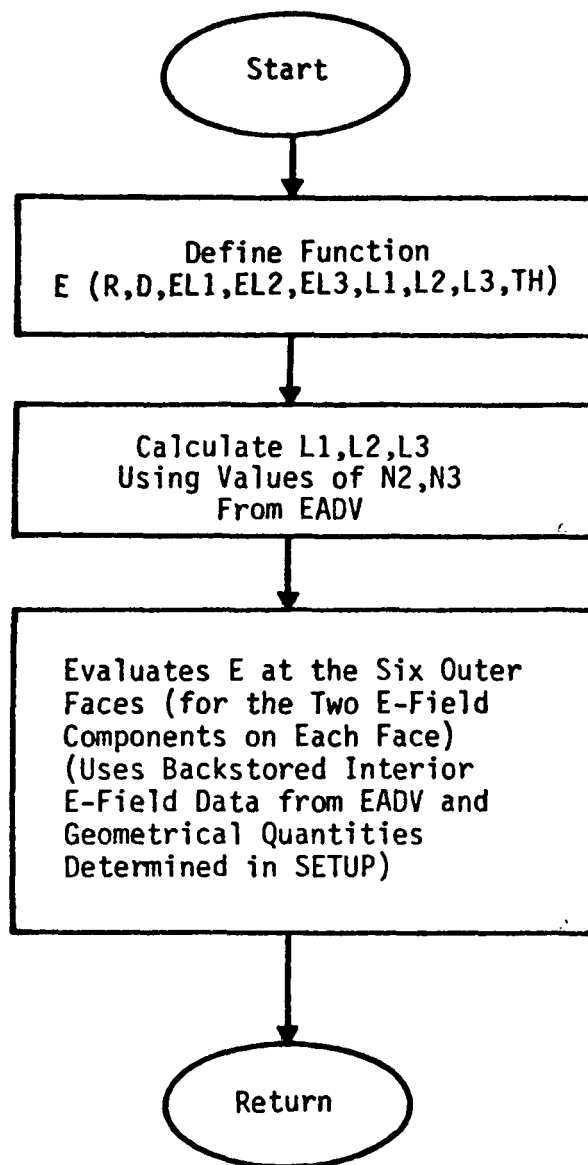


Figure 10. Flow Chart for Subroutine ERAD

SUBROUTINE ERAD

C

```
COMMON/EFIELD/EX(29,29,29),EY(29,29,29),EZ(29,29,29)
COMMON/GRID/X(28),Y(28),Z(28),X0(29),Y0(29),Z0(29),
1 DX(29),DY(29),DZ(29),DX0(28),DY0(28),DZ0(28),
2 DXI(29),DYI(29),DZI(29),DXOI(28),DYOI(28),DZOI(28)
COMMON/EXTRAS/NX,NY,NZ,NX1,NY1,NZ1,N,M,MQ,DT,XMU,EPSO,EPS,NPTLIM,
1 NN,NPTS,LMAX,SIGMA,C,T,PI,EXPFAC,IP,TX,TY,TZ,AMP,ALPHA,
2 BETA,IDLS
COMMON/RAD/RXY(28,29,2),RZX(29,28,2),RXY(28,29,2),RZY(29,28,2),
1 RXZ(28,29,2),RYZ(29,28,2),THYX(28,29,2),THZX(29,28,2),
2 THXY(28,29,2),THZY(29,28,2),THXZ(28,29,2),THYZ(29,28,2)
COMMON/EBS/EYXD(28,29,3),EYXU(28,29,3),EZXD(29,28,3),EZ XU(29,28,3)
1 ,EXYD(28,29,3),EXYU(28,29,3),EZYD(29,28,3),EZYU(29,28,3),
2 EXZD(28,29,3),EXZU(28,29,3),EYZD(29,28,3),EYZU(29,28,3),
3 N1,N2,N3
```

C

```
E(R,D,EL1,EL2,EL3,L1,L2,L3,TH) = R/D*(
1 EL1*(L2-L3)*(L2*L3+TH*(L2+L3)+TH**2)+
2 EL2*(L3-L1)*(L3*L1+TH*(L3+L1)+TH**2)+
3 EL3*(L1-L2)*(L1*L2+TH*(L1+L2)+TH**2))
```

C

C

```
L1=-1
L2=N-N2-1
L3=N-N3-1
D=L2*L3*(L2-L3)+L1*L3*(L3-L1)+L1*L2*(L1-L2)
IF(D.EQ.0.) D=1.E-9
```

C

C

C

RADIATE EXY

```
DO 11 I=1,NX1
DO 11 K=2,NZ1
EX(I,1,K)=E(RXY(I,K,1),D,EX(I,2,K),EXYD(I,K,2),EXYD(I,K,3),
1 L1,L2,L3,THXY(I,K,1))
EX(I,NY,K) =E(RXY(I,K,2),D,EX(I,NY1,K),EXYU(I,K,2),EXYU(I,K,3),
1 L1,L2,L3,THXY(I,K,2))
11 CONTINUE
```

C

C

C

RADIATE EXZ

```
DO 111 I=1,NX1
DO 111 J=2,NY1
EX(I,J,1)=E(RXZ(I,J,1),D,EX(I,J,2),EXZD(I,J,2),EXZD(I,J,3),
1 L1,L2,L3,THXZ(I,J,1))
EX(I,J,NZ) =E(RXZ(I,J,2),D,EX(I,J,NZ1),EXZU(I,J,2),EXZU(I,J,3),
1 L1,L2,L3,THXZ(I,J,2))
111 CONTINUE
```

C

C

C

RADIATE EYX

```
DO 22 J=1,NY1
DO 22 K=2,NZ1
EY(1,J,K)=E(RYX(J,K,1),D,EY(2,J,K),EYXD(J,K,2),EYXD(J,K,3),
1 L1,L2,L3,THYX(J,K,1))
```

```

      EY(NX,J,K) =E(RYX(J,K,2),D,EY(NX1,J,K),EYXU(J,K,2),EYXU(J,K,3),
1  L1,L2,L3,THYX(J,K,2))
22 CONTINUE
C
C  RADIATE EYZ
C
      DO 222 I=2,NX1
      DO 222 J=1,NY1
      EY(I,J,1)=E(RYZ(I,J,1),D,EY(I,J,2),EYZD(I,J,2),EYZD(I,J,3),
1  L1,L2,L3,THYZ(I,J,1))
      EY(I,J,NZ) =E(RYZ(I,J,2),D,EY(I,J,NZ1),EYZU(I,J,2),EYZU(I,J,3),
1  L1,L2,L3,THYZ(I,J,2))
222 CONTINUE
C
C  RADIATE EZX
C
      DO 33 J=2,NY1
      DO 33 K=1,NZ1
      EZ(1,J,K)=E(RZX(J,K,1),D,EZ(2,J,K),EZXD(J,K,2),EZXD(J,K,3),
1  L1,L2,L3,THZX(J,K,1))
      EZ(NX,J,K) =E(RZX(J,K,2),D,EZ(NX1,J,K),EZXU(J,K,2),EZXU(J,K,3),
1  L1,L2,L3,THZX(J,K,2))
33 CONTINUE
C
C  RADIATE EZY
C
      DO 333 I=2,NX1
      DO 333 K=1,NZ1
      EZ(I,1,K)=E(RZY(I,K,1),D,EZ(I,2,K),EZYD(I,K,2),EZYD(I,K,3),
1  L1,L2,L3,THZY(I,K,1))
      EZ(I,NY,K) =E(RZY(I,K,2),D,EZ(I,NY1,K),EZYU(I,K,2),EZYU(I,K,3),
1  L1,L2,L3,THZY(I,K,2))
333 CONTINUE
C
      RETURN
      END

```

3.5.6 Subroutine HADV

In subroutine HADV, the equation

$$\nabla \times E = -\mu \frac{\partial H}{\partial t} \quad (16)$$

is applied to the scattered fields and is used to find H from E and H at earlier times. This "time stepping" solution is found for each component of H, i.e., Equation (16) is written explicitly for each component as

$$-\mu \frac{\partial H_x^s}{\partial t} = \frac{\partial E_z^s}{\partial y} - \frac{\partial E_y^s}{\partial z}$$

$$-\mu \frac{\partial H_y^s}{\partial t} = \frac{\partial E_x^s}{\partial z} - \frac{\partial E_z^s}{\partial x}$$

$$-\mu \frac{\partial H_z^s}{\partial t} = \frac{\partial E_x^s}{\partial y} - \frac{\partial E_y^s}{\partial x}$$

These equations are cast in their finite-difference form and solved for H at the latest time. Thus

$$H_x^n(I, J, K) = H_x^{n-1}(I, J, K) - (\Delta t / \mu) \left(\frac{E_z^{n-1/2}(I, J+1, K) - E_z^{n-1/2}(I, J, K)}{Y_o(J+1) - Y_o(J)} - \frac{E_y^{n-1/2}(I, J, K+1) - E_y^{n-1/2}(I, J, K)}{Z_o(K+1) - Z_o(K)} \right) \quad (17)$$

$$H_y^n(I, J, K) = H_y^{n-1}(I, J, K) - (\Delta t / \mu) \left(\frac{E_x^{n-1/2}(I, J, K+1) - E_x^{n-1/2}(I, J, K)}{Z_o(K+1) - Z_o(K)} - \frac{E_z^{n-1/2}(I+1, J, K) - E_z^{n-1/2}(I, J, K)}{X_o(I+1) - X_o(I)} \right) \quad (18)$$

$$H_z^n(I, J, K) = H_z^{n-1}(I, J, K) - (\Delta t / \mu) \left(\frac{E_y^{n-1/2}(I+1, J, K) - E_y^{n-1/2}(I, J, K)}{X_o(I+1) - X_o(I)} - \frac{E_x^{n-1/2}(I, J+1, K) - E_x^{n-1/2}(I, J, K)}{Y_o(J+1) - Y_o(J)} \right) \quad (19)$$

It is assumed that $\mu = \mu_o$ for all regions of the problem space.

HADV is called every program cycle by DRIVER to advance the H-fields, just as Subroutine EADV is called every program cycle to advance the E-fields. It uses the prior values of the fields to calculate the new values. Initially, these prior values are zero. The first nonzero values are those values for E. These values are accessed by a call to EBC in the Subroutine EADV for advancing the E-fields. Thus, on the next program cycle these values will be available and nonzero values of the fields off the boundary will start appearing when Equations (17), (18), and (19) are advanced again. Since EBC is always called after the fields are advanced, the proper boundary conditions are maintained.

HADV also defines a loop of H fields that drives the lightning channel for $M=1$.

The only variable used locally in HADV is DTMU, which is equal to $\Delta t / \mu$. Flow for the subroutine is shown in Figure 11.

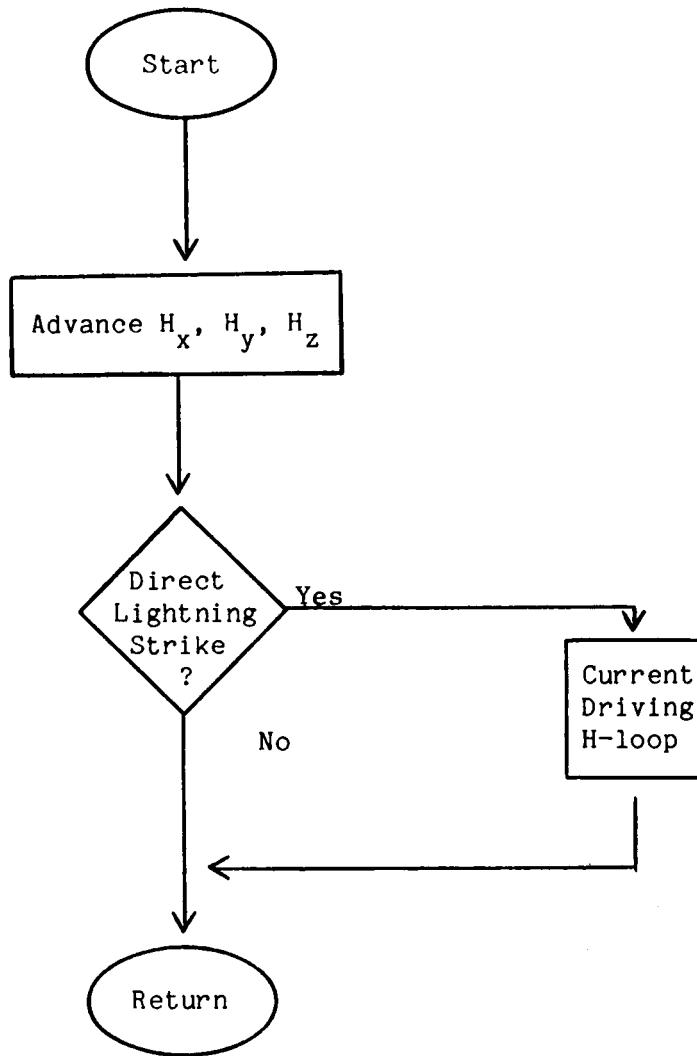


Figure 11. Flow Chart for Subroutine HADV

C-2

SUBROUTINE HADV

```

C
COMMON/EFIELD/EX(29,29,29),EY(29,29,29),EZ(29,29,29)
COMMON/HFIELD/HX(29,29,29),HY(29,29,29),HZ(29,29,29)
COMMON/GRID/X(28),Y(28),Z(28),XO(29),YO(29),ZO(29),
1  LX(29),DY(29),DL(29),DXO(28),DYO(28),DZO(28),
2  DXI(29),DYI(29),DZI(29),DXOI(28),DYOI(28),DZOI(28)
COMMON/EXTRAS/NX,NY,NZ,NX1,NY1,NZ1,N,M,MQ,DT,XMU,EPSO,EPS,NPTLIM,
1  NN,NPTS,LMAX,SIGMA,C,T,PI,EXPFAC,IP,TX,TY,TZ,AMP,ALPHA,BETA,IDLS

C      DTXMU = DT/XMU

C
C      ADVANCE HX
C
      DO 1 I = 2,NX1
      DO 1 J = 1,NY1
      DO 1 K = 1,NZ1
      HX(I,J,K) = HX(I,J,K) - DTXMU*((EZ(I,J+1,K) - EZ(I,J,K))*DYOI(J)
1  - (EY(I,J,K+1) - EY(I,J,K))*DZOI(K))
1  CONTINUE

C
C      ADVANCE HY
C
      DO 2 I = 1,NX1
      DO 2 J = 2,NY1
      DO 2 K = 1,NZ1
      HY(I,J,K) = HY(I,J,K) - DTXMU*((EX(I,J,K+1) - EX(I,J,K))*DZOI(K)
1  - (EZ(I+1,J,K) - EZ(I,J,K))*DXOI(I))
2  CONTINUE

C
C      ADVANCE HZ
C
      DO 3 I = 1,NX1
      DO 3 J = 1,NY1
      DO 3 K = 2,NZ1
      HZ(I,J,K) = HZ(I,J,K) - DTXMU*((EY(I+1,J,K) - EY(I,J,K))*DXOI(I)
1  - (EX(I,J+1,K) - EX(I,J,K))*DYOI(J))
3  CONTINUE

C
C
C      IF(IDLS.EQ.0) GO TO 1

C
C      IF(M.NE.1) GO TO 1

C
C      M=1 LOOP,H-LOOP SOURCE

C
C      M=2 LOOP NOT TREATED-SOURCE MUST BE OUTSIDE THE PROBLEM SPACE
C      AND THE WIRES ARE ALSO OUTSIDE

C
      EXPAB=EXP(-ALPHA*T)-EXP(-BETA*T)

C
      HX(8,21,15)=-((33333.)*EXPAB
      HZ(8,21,15)=(16667.)*EXPAB
      HX(8,21,14)=(33333.)*EXPAB
      HZ(7,21,15)=-((16667.)*EXPAB

```

C
1 CONTINUE

C
C

RETURN
END

3.5.7 FIELDS

There are nine functions* used for calculating the six incident E- and H-field components: EINCX, EINCX, EINCZ, EINCXP, EINCYP, EINCZP, HINCXP, HINCYP, and HINCZP. The first letter in the function, E or H, tells what field; the next three letters, INC, indicate that these are incident field components; the next letter, X, Y or Z, states what component is evaluated at a point in space located at XP, YP, ZP and if P is not present the field component is evaluated at the field component location in the cell space labeled by I,J,K. EINCX, EINCX, and EINCZ are called from Subroutine EBC. The others are called by Subroutine DATASAV.

Two different sources for the incident fields are considered:

- 1) direct strikes;
- 2) indirect illumination

Direct strikes are modeled by H-fields impressed around a perfectly conducting wire leading to the aircraft that represents the lightning channel. If a lightning channel resistance is desired, then use the method outlined in Appendix D, in the discussion on excitation sources. The specified H-fields determine the lightning channel's current time history. Indirect illumination is modeled by the fields directly illuminating the aircraft, such as EINCX and HINCZ broadside illumination from above.

* While these functions are not subroutines per se, they are similar enough in nature to warrant inclusion in this section, as opposed to having a separate short section of their own.

```

FUNCTION EINCX(I,J,K)
C
COMMON/UGRID/UX(28),UY(28),UZ(28),UXO(29),UYO(29),UZO(29)
COMMON/GRID/X(28),Y(28),Z(28),XO(29),YO(29),ZO(29),
1 DX(29),DY(29),DZ(29),DXO(28),DYO(28),DZO(28),
2 DXI(29),DYI(29),DZI(29),DXOI(28),DYOI(28),DZOI(28)
COMMON/EXTRAS/NX,NY,NZ,NX1,NY1,NZ1,N,M,MQ,DT,XMU,EPSO,EPS,NPTLIM,
1 NN,NPTS,LMAX,SIGMA,C,T,PI,EXPFAC,IP,TX,TY,TZ,AMP,ALPHA,BETA,IDLS
C
C      THREAT
C
EINCX = 0.0
IF(IDLS.EQ.1) GO TO 1
TAU = T - (UYO(23) - YO(J))/C
IF (TAU.LE.0.0) GO TO 1
EINCX=AMP*(EXP(-ALPHA*TAU)-EXP(-BETA*TAU))
1 CONTINUE
RETURN
END
C
FUNCTION DEINCX(I,J,K)
C
COMMON/UGRID/UX(28),UY(28),UZ(28),UXO(29),UYO(29),UZO(29)
COMMON/GRID/X(28),Y(28),Z(28),XO(29),YO(29),ZO(29),
1 DX(29),DY(29),DZ(29),DXO(28),DYO(28),DZO(28),
2 DXI(29),DYI(29),DZI(29),DXOI(28),DYOI(28),DZOI(28)
COMMON/EXTRAS/NX,NY,NZ,NX1,NY1,NZ1,N,M,MQ,DT,XMU,EPSO,EPS,NPTLIM,
1 NN,NPTS,LMAX,SIGMA,C,T,PI,EXPFAC,IP,TX,TY,TZ,AMP,ALPHA,BETA,IDLS
DEINCX=0.0
IF(IDLS.EQ.1) GO TO 1
TAU=T-(UYO(23)-YO(J))/C
IF(TAU.LE.0) GO TO 1
DEINCX=AMP*(BETA*(EXP(-BETA*TAU))-ALPHA*(EXP(-ALPHA*TAU)))
1 CONTINUE
RETURN
END
C
FUNCTION EINCY(I,J,K)
C
COMMON/UGRID/UX(28),UY(28),UZ(28),UXO(29),UYO(29),UZO(29)
COMMON/GRID/X(28),Y(28),Z(28),XO(29),YO(29),ZO(29),
1 DX(29),DY(29),DZ(29),DXO(28),DYO(28),DZO(28),
2 DXI(29),DYI(29),DZI(29),DXOI(28),DYOI(28),DZOI(28)
COMMON/EXTRAS/NX,NY,NZ,NX1,NY1,NZ1,N,M,MQ,DT,XMU,EPSO,EPS,NPTLIM,
1 NN,NPTS,LMAX,SIGMA,C,T,PI,EXPFAC,IP,TX,TY,TZ,AMP,ALPHA,BETA,IDLS
C
C      THREAT
C
EINCY = 0.0
RETURN
END
C
FUNCTION DEINCY(I,J,K)
DEINCY=0.0
RETURN

```

```

END
C
FUNCTION EINCZ(I,J,K)
C
COMMON/UGRID/UX(28),UY(28),UZ(28),UXO(29),UYO(29),UZO(29)
COMMON/GRID/X(28),Y(28),Z(28),XO(29),YO(29),ZO(29),
1 DX(29),DY(29),DZ(29),DXO(28),DYO(28),DZO(28),
2 DXI(29),DYI(29),DZI(29),DXOI(28),DYOI(28),DZOI(28)
COMMON/EXTRAS/NX,NY,NZ,NX1,NY1,NZ1,N,M,MQ,DT,XMU,EPSO,EPS,NPTLIM,
1 NN,NPTS,LMAX,SIGMA,C,T,PI,EXPFAC,IP,TX,TY,TZ,AMP,ALPHA,BETA,IDLS
C
C      THREAT
C
EINCZ = 0.0
RETURN
END
C
FUNCTION DEINCZ(I,J,K)
C
COMMON/UGRID/UX(28),UY(28),UZ(28),UXO(29),UYO(29),UZO(29)
COMMON/GRID/X(28),Y(28),Z(28),XO(29),YO(29),ZO(29),
1 DX(29),DY(29),DZ(29),DXO(28),DYO(28),DZO(28),
2 DXI(29),DYI(29),DZI(29),DXOI(28),DYOI(28),DZOI(28)
COMMON/EXTRAS/NX,NY,NZ,NX1,NY1,NZ1,N,M,MQ,DT,XMU,EPSO,EPS,NPTLIM,
1 NN,NPTS,LMAX,SIGMA,C,T,PI,EXPFAC,IP,TX,TY,TZ,AMP,ALPHA,BETA,IDLS
DEINCZ=0.0
RETURN
END
C
FUNCTION HINCX(I,J,K)
C
COMMON/UGRID/UX(28),UY(28),UZ(28),UXO(29),UYO(29),UZO(29)
COMMON/GRID/X(28),Y(28),Z(28),XO(29),YO(29),ZO(29),
1 DX(29),DY(29),DZ(29),DXO(28),DYO(28),DZO(28),
2 DXI(29),DYI(29),DZI(29),DXOI(28),DYOI(28),DZOI(28)
COMMON/EXTRAS/NX,NY,NZ,NX1,NY1,NZ1,N,M,MQ,DT,XMU,EPSO,EPS,NPTLIM,
1 NN,NPTS,LMAX,SIGMA,C,T,PI,EXPFAC,IP,TX,TY,TZ,AMP,ALPHA,BETA,IDLS
C
C      THREAT
C
HINCX=0.0
RETURN
END
C
FUNCTION HINCY(I,J,K)
C
COMMON/UGRID/UX(28),UY(28),UZ(28),UXO(29),UYO(29),UZO(29)
COMMON/GRID/X(28),Y(28),Z(28),XO(29),YO(29),ZO(29),
1 DX(29),DY(29),DZ(29),DXO(28),DYO(28),DZO(28),
2 DXI(29),DYI(29),DZI(29),DXOI(28),DYOI(28),DZOI(28)
COMMON/EXTRAS/NX,NY,NZ,NX1,NY1,NZ1,N,M,MQ,DT,XMU,EPSO,EPS,NPTLIM,
1 NN,NPTS,LMAX,SIGMA,C,T,PI,EXPFAC,IP,TX,TY,TZ,AMP,ALPHA,BETA,IDLS
C
C      THREAT
C

```

```

HINCY=0.0
RETURN
END

C
FUNCTION HINCZ(I,J,K)
C
COMMON/UGRID/UX(28),UY(28),UZ(28),UX0(29),UY0(29),UZ0(29)
COMMON/GRID/X(28),Y(28),Z(28),X0(29),Y0(29),Z0(29),
1 DX(29),DY(29),DZ(29),DX0(28),DY0(28),DZ0(28),
2 DXI(29),DYI(29),DZI(29),DXOI(28),DYOI(28),DZOI(28)
COMMON/EXTRAS/NX,NY,NZ,NX1,NY1,NZ1,N,M,MQ,DT,XMU,EPSO,EPS,NPTLIM,
1 NN,NPTS,LMAX,SIGMA,C,T,PI,EXPFAC,IP,TX,TY,TZ,AMP,ALPHA,BETA,IDLS

C
C THREAT
C
HINCZ=0.0
RETURN
END

C
FUNCTION EINCXP(XP,YP,ZP)
C
COMMON/UGRID/UX(28),UY(28),UZ(28),UX0(29),UY0(29),UZ0(29)
COMMON/GRID/X(28),Y(28),Z(28),X0(29),Y0(29),Z0(29),
1 DX(29),DY(29),DZ(29),DX0(28),DY0(28),DZ0(28),
2 DXI(29),DYI(29),DZI(29),DXOI(28),DYOI(28),DZOI(28)
COMMON/EXTRAS/NX,NY,NZ,NX1,NY1,NZ1,N,M,MQ,DT,XMU,EPSO,EPS,NPTLIM,
1 NN,NPTS,LMAX,SIGMA,C,T,PI,EXPFAC,IP,TX,TY,TZ,AMP,ALPHA,BETA,IDLS

C
C THREAT
C
EINCXP = 0.0
IF(IDLS.EQ.1) GO TO 1
TAU = T - (UY0(23) - YP)/C
IF (TAU.LE.0) GO TO 1
EINCXP = AMP * (EXP(-ALPHA*TAU) - EXP(-BETA*TAU))
1 CONTINUE
RETURN
END

C
FUNCTION EINCYP(XP,YP,ZP)
C
COMMON/UGRID/UX(28),UY(28),UZ(28),UX0(29),UY0(29),UZ0(29)
COMMON/GRID/X(28),Y(28),Z(28),X0(29),Y0(29),Z0(29),
1 DX(29),DY(29),DZ(29),DX0(28),DY0(28),DZ0(28),
2 DXI(29),DYI(29),DZI(29),DXOI(28),DYOI(28),DZOI(28)
COMMON/EXTRAS/NX,NY,NZ,NX1,NY1,NZ1,N,M,MQ,DT,XMU,EPSO,EPS,NPTLIM,
1 NN,NPTS,LMAX,SIGMA,C,T,PI,EXPFAC,IP,TX,TY,TZ,AMP,ALPHA,BETA,IDLS

C
EINCYP = 0.0
RETURN
END

C
FUNCTION EINCZP(XP,YP,ZP)
C
COMMON/UGRID/UX(28),UY(28),UZ(28),UX0(29),UY0(29),UZ0(29)

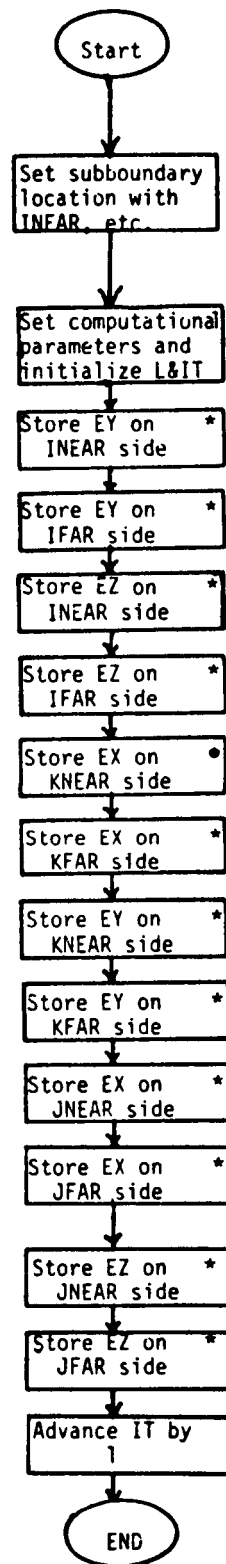
```


3.5.8 Subroutine SAVESB

The subroutine SAVESB saves the tangential E-field components on the subboundary selected for the M=1 loop, when the model is unexpanded. The subboundary is defined by the values set in a data statement for INEAR, JNEAR, KNEAR, IFAR, JFAR, KFAR. The values must be consistent with the expansion factor value selected. For example, an expansion factor of 4 (2,4,7 and 14 are allowed for a problem space 28 cells on a side) require seven cells on a side in the subboundary volume. In this case the values of INEAR and IFAR would be N and N+7, where N is an integer from 1 to 22.

SAVESB treats opposing sides of the subboundary in succession (INEAR and IFAR, JNEAR and JFAR, KNEAR and KFAR sides respectively), saving for each side the two E-field components tangential to the surface (EY and EZ for the INEAR and IFAR sides for example) in the ARAY (L, IT). These components on a seven cell on a side subboundary form 9x8 and 8x9 arrays on a side. This provides field components on the edges of the subboundary (8 field components field components on each cell edge) or bracketing the edges of the subboundary (9 field components across and one cell on either side of the face at each cell center). As a result, in OUTBND it is possible to interpolate field values between the ARAY stored values using a single simple interpolation scheme. If the components running 9 across and about a face were not saved, rather only 7 across the face, then an extrapolation scheme for the fields along the edges would be required, unnecessarily complicating matters.

With 9x8 and 8x9 arrays of two field components on a face and 6 faces, the total number of stored components for a single call to SAVESB is 864, when an expansion factor of 4 is selected. SAVESB is called for each time step, which is labeled by IT in SAVESB, and ARAY is dimensioned as ARAY (864,500), when the expansion factor is 4 and no more than 500 time steps are required in the M=1 loop. These dimensions must be manually reset for other cases. Figure 13 is a flow chart of SAVESB.



• L is advanced by one for every stored value

Figure 12. SAVESB Flow Chart

```

SUBROUTINE SAVESB
C
COMMON/EFIELD/EX(29,29,29),EY(29,29,29),EZ(29,29,29)
COMMON/EXTRAS/NX,NY,NZ,NX1,NY1,NZ1,N,M,MQ,DT,XMU,EPSO,EPS,NPTLIM,
1 NN,NPTS,LMAX,SIGMA,C,T,PI,EXPFAC,IP,TX,TY,TZ,AMP,ALPHA,BETA,IDLS
C
COMMON/RAY/ARAY(864,500)
C
DATA INEAR,JNEAR,KNEAR,IFAR,JFAR,KFAR/7,9,12,14,16,19/
C
THIS SUBROUTINE SAVES THE TANGENTIAL E FIELD COMPONENTS ON THE
SUBBOUNDARY
C
EXPFAC=4.0
LIMIT=28/EXPFAC
L=1
IF(N.NE.0) GO TO 10
IT=1
PRINT 5
5 FORMAT(*SAVESB CALLED*)
10 CONTINUE
C
IMIN=INEAR
JMIN=JNEAR
KMIN=KNEAR
C
EY ON INEAR,J,K SIDE
C
IMAX=IMIN+LIMIT
JMAX=JMIN+LIMIT
KMAX=KMIN+LIMIT
C
IL=IMIN-1
JL=JMIN-1
KL=KMIN-1
C
DO 90 J=JL,JMAX
DO 90 K=KMIN,KMAX
ARAY(L,IT)=EY(INEAR,J,K)
L=L+1
90 CONTINUE
C
EY ON IFAR,J,K SIDE
C
DO 110 J=JL,JMAX
DO 110 K=KMIN,KMAX
ARAY(L,IT)=EY(IFAR,J,K)
L=L+1
110 CONTINUE
C
EZ ON INEAR,J,K SIDE
C
DO 130 J=JMIN,JMAX
DO 130 K=KL,KMAX

```

```

        ARAY(L,IT)=EZ(INEAR,J,K)
        L=L+1
130 CONTINUE
C
C     EZ ON IFAR,J,K SIDE
C
        DO 150 J=JMIN,JMAX
        DO 150 K=KL,KMAX
        ARAY(L,IT)=EZ(IFAR,J,K)
        L=L+1
150 CONTINUE
C
C     EX ON I,J,KNEAR SIDE
C
C
C
        DO 170 I=IL,IMAX
        DO 170 J=JMIN,JMAX
        ARAY(L,IT)=EX(I,J,KNEAR)
        L=L+1
170 CONTINUE
C
C     EX ON I,J,KFAR SIDE
C
C
        DO 190 I=IL,IMAX
        DO 190 J=JMIN,JMAX
        ARAY(L,IT)=EX(I,J,KFAR)
        L=L+1
190 CONTINUE
C
C     EY ON I,J,KNEAR SIDE
C
C
C
        DO 210 I=IMIN,IMAX
        DO 210 J=JL,JMAX
        ARAY(L,IT)=EY(I,J,KNEAR)
        L=L+1
210 CONTINUE
C
C     EY ON I,J,KFAR SIDE
C
C
        DO 230 I=IMIN,IMAX
        DO 230 J=JL,JMAX
        ARAY(L,IT)=EY(I,J,KFAR)
        L=L+1
230 CONTINUE
C
C     EX ON I,JNEAR,K SIDE
C
C
C
        DO 250 I=IL,IMAX
        DO 250 K=KMIN,KMAX
        ARAY(L,IT)=EX(I,JNEAR,K)
        L=L+1
250 CONTINUE
C

```

```

C      EX ON I,JFAR,K SIDE
C
      DO 270 I=IL,IMAX
      DO 270 K=KMIN,KMAX
      ARAY(L,IT)=EX(I,JFAR,K)
      L=L+1
270 CONTINUE
C
C      EZ ON I,JNEAR,K SIDE
C
C
      DO 290 I=IMIN,IMAX
      DO 290 K=KL,KMAX
      ARAY(L,IT)=EZ(I,JNEAR,K)
      L=L+1
290 CONTINUE
C
C      C      EZ ON I,JFAR,K SIDE
C
      DO 310 I=IMIN,IMAX
      DO 310 K=KL,KMAX
      ARAY(L,IT)=EZ(I,JFAR,K)
      L=L+1
      IF(IT.NE.1) GO TO 310
310 CONTINUE
C
      IT=IT+1
      RETURN
      END

```

3.5.9 Subroutine OUTBND

Subroutine OUTBND in the M=2 loop reads the data from ARAY acquired in SAVESB in the M=1 loop via a common statement. The data is first interpolated in time with each time step replaced by an expansion factor (EXPFAC or as in the present code 4) of time steps. Next the data is interpolated spatially between the unexpanded grid points (UX, UY, UZ, UX0, UY0, UZ0 of COMMON/UGRID/...) with a simple linear scheme and the expanded grid point locations (X,Y,Z,X0,Y0,Z0 of COMMON/GRID/...). The interpolated data is placed on the faces of the problem space in the M=2 loop and acts in lieu of the radiation boundary condition defined in ERAD that is used on the M=1 loop. The same sequence of faces and components as used in SAVESB is used in OUTBND to insure that the interpolated data appears on the correct face in the correct location. Figure 14 is a flow chart of OUTBND.

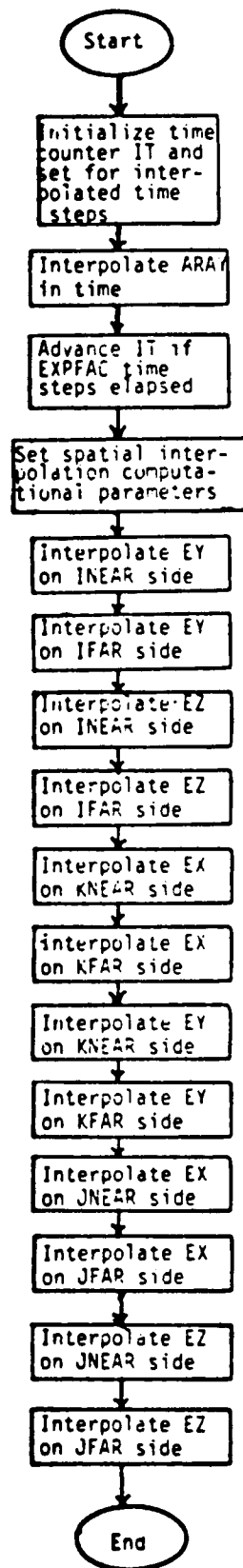


Figure 13. OUTBND Flow Chart

```

SUBROUTINE OUTBND
C
C THIS SUBROUTINE ESTABLISHES THE TANGENTIAL E-FIELD COMPONENTS
C ALONG THE OUTER BOUNDARY OF THE SUBBOUNDARY
C
COMMON/EFIELD/EX(29,29,29),EY(29,29,29),EZ(29,29,29)
COMMON/GRID/X(28),Y(28),Z(28),X0(29),Y0(29),Z0(29),
1 DX(29),DY(29),DZ(29),DX0(28),DY0(28),DZ0(28),
2 DXI(29),DYI(29),DZI(29),DXOI(28),DYOI(28),DZOI(28)
COMMON/UGRID/UX(28),UY(28),UZ(28),UX0(29),UY0(29),UZ0(29)
COMMON/EXTRAS/NX,NY,NZ,NX1,NY1,NZ1,N,M,MQ,DT,XMU,EPSO,EPS,NPTLIM,
1 NN,NPTS,LMAX,SIGMA,C,T,PI,EXPFAC,IP,TX,TY,TZ,AMP,ALPHA,BETA,IDLS
C
COMMON/RAY/ARRAY(864,500)
DIMENSION ARRAY(864),CRAY(9,9)
C
DATA INEAR,JNEAR,KNEAR,IFAR,JFAR,KFAR/7,9,12,14,16,19/
IF(N.NE.0) GO TO 10
PRINT 5
5 FORMAT(*OUTBND CALLED*)
IT=1
10 CONTINUE
E=MOD(N,4)
C
DO 50 LL=1,864
ARRAY(LL)=ARRAY(LL,IT)+(E/EXPFAC)*(ARRAY(LL,IT+1)-ARRAY(LL,IT))
50 CONTINUE
C
IF(E.NE.3) GO TO 20
IT=IT+1
20 CONTINUE
C
LIMIT=28/EXPFAC
LONE=LIMIT+1
LTWO=LIMIT+2
LL=1
DO 80 J=1,LTWO
DO 80 K=1,LONE
CRAY(J,K)=ARRAY(LL)
LL=LL+1
80 CONTINUE
C
DO 90 J=1,28
JJ=INT((J+1)/EXPFAC)+1
DO 90 K=1,28
KK=INT((K-2)/EXPFAC)+1
JS=JJ+JNEAR-1
KS=KK+KNEAR-1
EY1=CRAY(JJ+1,KK)+((Z0(K)-UZ0(KS))/(UZ0(KS+1)-UZ0(KS)))
1 *(CRAY(JJ+1,KK+1)-CRAY(JJ+1,KK))
EY2=CRAY(JJ,KK)+((Z0(K)-UZ0(KS))/(UZ0(KS+1)-UZ0(KS)))
1 *(CRAY(JJ,KK+1)-CRAY(JJ,KK))
EY(1,J,K)=EY2+((Y(J)-UY(JS-1))/(UY(JS)-UY(JS-1)))
1 *(EY1-EY2)
90 CONTINUE

```



```

C
DO 100 J=1,LTWO
DO 100 K=1,LONE
CRAY(J,K)=ARRAY(LL)
LL=LL+1
100 CONTINUE
C
DO 110 J=1,28
JJ=INT((J+1)/EXPFAC)+1
DO 110 K=1,28
KK=INT((K-2)/EXPFAC)+1
JS=JJ+JNEAR-1
KS=KK+KNEAR-1
EY1=CRAY(JJ+1,KK)+((ZO(K)-UZO(KS))/(UZO(KS+1)-UZO(KS)))
1      *(CRAY(JJ+1,KK+1)-CRAY(JJ+1,KK))
EY2=CRAY(JJ,KK)+((ZO(K)-UZO(KS))/(UZO(KS+1)-UZO(KS)))
1      *(CRAY(JJ,KK+1)-CRAY(JJ,KK))
EY(29,J,K)=EY2+((Y(J)-UY(JS-1))/(UY(JS)-UY(JS-1)))
1      *(EY1-EY2)
110 CONTINUE
C
DO 120 J=1,LONE
DO 120 K=1,LTWO
CRAY(J,K)=ARRAY(LL)
LL=LL+1
120 CONTINUE
C
DO 130 J=1,28
JJ=INT((J-2)/EXPFAC)+1
DO 130 K=1,28
KK=INT((K+1)/EXPFAC)+1
JS=JJ+JNEAR-1
KS=KK+KNEAR-1
EZ1=CRAY(JJ+1,KK)+((Z(K)-UZ(KS-1))/(UZ(KS)-UZ(KS-1)))
1      *(CRAY(JJ+1,KK+1)-CRAY(JJ+1,KK))
EZ2=CRAY(JJ,KK)+((Z(K)-UZ(KS-1))/(UZ(KS)-UZ(KS-1)))
1      *(CRAY(JJ,KK+1)-CRAY(JJ,KK))
EZ(1,J,K)=EZ2+((YO(J)-UYO(JS))/(UYO(JS+1)-UYO(JS)))
1      *(EZ1-EZ2)
130 CONTINUE
C
DO 140 J=1,LONE
DO 140 K=1,LTWO
CRAY(J,K)=ARRAY(LL)
LL=LL+1
140 CONTINUE
C
DO 150 J=1,28
JJ=INT((J-2)/EXPFAC)+1
DO 150 K=1,28
KK=INT((K+1)/EXPFAC)+1
JS=JJ+JNEAR-1
KS=KK+KNEAR-1
EZ1=CRAY(JJ+1,KK)+((Z(K)-UZ(KS-1))/(UZ(KS)-UZ(KS-1)))
1      *(CRAY(JJ+1,KK+1)-CRAY(JJ+1,KK))

```

```

      EZ2=CRAY(JJ,KK)+((Z(K)-UZ(KS-1))/(UZ(KS)-UZ(KS-1)))
1      *(CRAY(JJ,KK+1)-CRAY(JJ,KK))
      EZ(29,J,K)=EZ2+((YO(J)-UYO(JS))/(UYO(JS+1)-UYO(JS)))
1      *(EZ1-EZ2)
150 CONTINUE
C
      DO 160 I=1,LTWO
      DO 160 J=1,LONE
      CRAY(I,J)=ARRAY(LL)
      LL=LL+1
160 CONTINUE
C
      DO 170 I=1,28
      II=INT((I+1)/EXPFAC)+1
      DO 170 J=1,28
      JJ=INT((J-2)/EXPFAC)+1
      IS=II+INEAR-1
      JS=JJ+JNEAR-1
      EX1=CRAY(II,JJ+1)+((X(I)-UX(IS-1))/(UX(IS)-UX(IS-1)))
1      *(CRAY(II+1,JJ+1)-CRAY(II,JJ+1))
      EX2=CRAY(II,JJ)+((X(I)-UX(IS-1))/(UX(IS)-UX(IS-1)))
1      *(CRAY(II+1,JJ)-CRAY(II,JJ))
      EX(I,J,1)=EX2+((YO(J)-UYO(JS))/(UYO(JS+1)-UYO(JS)))
1      *(EX1-EX2)
170 CONTINUE
C
      DO 180 I=1,LTWO
      DO 180 J=1,LONE
      CRAY(I,J)=ARRAY(LL)
      LL=LL+1
180 CONTINUE
C
      DO 190 I=1,28
      II=INT((I+1)/EXPFAC)+1
      DO 190 J=1,28
      JJ=INT((J-2)/EXPFAC)+1
      IS=II+INEAR-1
      JS=JJ+JNEAR-1
      EX1=CRAY(II,JJ+1)+((X(I)-UX(IS-1))/(UX(IS)-UX(IS-1)))
1      *(CRAY(II+1,JJ+1)-CRAY(II,JJ+1))
      EX2=CRAY(II,JJ)+((X(I)-UX(IS-1))/(UX(IS)-UX(IS-1)))
1      *(CRAY(II+1,JJ)-CRAY(II,JJ))
      EX(I,J,29)=EX2+((YO(J)-UYO(JS))/(UYO(JS+1)-UYO(JS)))
1      *(EX1-EX2)
190 CONTINUE
C
      DO 200 I=1,LONE
      DO 200 J=1,LTWO
      CRAY(I,J)=ARRAY(LL)
      LL=LL+1
200 CONTINUE
C
      DO 210 I=1,28
      II=INT((I-2)/EXPFAC)+1
      DO 210 J=1,28

```

```

      JJ=INT((J+1)/EXPFAC)+1
      IS=II+INEAR-1
      JS=JJ+JNEAR-1
      EY1=CRAY(II,JJ+1)+((XO(I)-UXO(IS))/(UXO(IS+1)-UXO(IS)))
1      *(CRAY(II+1,JJ+1)-CRAY(II,JJ+1))
      EY2=CRAY(II,JJ)+((XO(I)-UXO(IS))/(UXO(IS+1)-UXO(IS)))
1      *(CRAY(II+1,JJ)-CRAY(II,JJ))
      EY(I,J,1)=EY2+((Y(J)-UY(JS-1))/(UY(JS)-UY(JS-1)))
1      *(EY1-EY2)
210 CONTINUE
C
      DO 220 I=1,LONE
      DO 220 J=1,LTWO
      CRAY(I,J)=ARRAY(LL)
      LL=LL+1
220 CONTINUE
C
      DO 230 I=1,28
      II=INT((I-2)/EXPFAC)+1
      DO 230 J=1,28
      JJ=INT((J+1)/EXPFAC)+1
      IS=II+INEAR-1
      JS=JJ+JNEAR-1
      EY1=CRAY(II,JJ+1)+((XO(I)-UXO(IS))/(UXO(IS+1)-UXO(IS)))
1      *(CRAY(II+1,JJ+1)-CRAY(II,JJ+1))
      EY2=CRAY(II,JJ)+((XO(I)-UXO(IS))/(UXO(IS+1)-UXO(IS)))
1      *(CRAY(II+1,JJ)-CRAY(II,JJ))
      EY(I,J,29)=EY2+((Y(J)-UY(JS-1))/(UY(JS)-UY(JS-1)))
1      *(EY1-EY2)
230 CONTINUE
C
      DO 240 I=1,LTWO
      DO 240 K=1,LONE
      CRAY(I,K)=ARRAY(LL)
      LL=LL+1
240 CONTINUE
C
      DO 250 I=1,28
      II=INT((I+1)/EXPFAC)+1
      DO 250 K=1,28
      KK=INT((K-2)/EXPFAC)+1
      IS=II+INEAR-1
      KS=KK+KNEAR-1
      EX1=CRAY(II,KK+1)+((X(I)-UX(IS-1))/(UX(IS)-UX(IS-1)))
1      *(CRAY(II+1,KK+1)-CRAY(II,KK+1))
      EX2=CRAY(II,KK)+((X(I)-UX(IS-1))/(UX(IS)-UX(IS-1)))
1      *(CRAY(II+1,KK)-CRAY(II,KK))
      EX(I,1,K)=EX2+((ZO(K)-UZO(KS))/(UZO(KS+1)-UZO(KS)))
1      *(EX1-EX2)
C
250 CONTINUE
      DO 260 I=1,LTWO
      DO 260 K=1,LONE
      CRAY(I,K)=ARRAY(LL)
      LL=LL+1

```

260 CONTINUE

C

```
DO 270 I=1,28
  II=INT((I+1)/EXPFAC)+1
DO 270 K=1,28
  KK=INT((K-2)/EXPFAC)+1
  IS=II+INEAR-1
  KS=KK+KNEAR-1
  EX1=CRAY(II, KK+1)+((X(I)-UX(IS-1))/(UX(IS)-UX(IS-1)))
1*(CRAY(II+1, KK+1)-CRAY(II, KK+1))
  EX2=CRAY(II, KK)+((X(I)-UX(IS-1))/(UX(IS)-UX(IS-1)))
1*(CRAY(II+1, KK)-CRAY(II, KK))
  EX(I, 29, K)=EX2+((ZO(K)-UZO(KS))/(UZO(KS+1)-UZO(KS)))
1*(EX1-EX2)
```

270 CONTINUE

C

```
DO 280 I=1, LONE
DO 280 K=1, LTWO
  CRAY(I, K)=ARRAY(LL)
  LL=LL+1
```

280 CONTINUE

C

```
DO 290 I=1,28
  II=INT((I-2)/EXPFAC)+1
DO 290 K=1,28
  KK=INT((K+1)/EXPFAC)+1
  IS=II+INEAR-1
  KS=KK+KNEAR-1
  EZ1=CRAY(II, KK+1)+((XO(I)-UXO(IS))/(UXO(IS+1)-UXO(IS)))
1*(CRAY(II+1, KK+1)-CRAY(II, KK+1))
  EZ2=CRAY(II, KK)+((XO(I)-UXO(IS))/(UXO(IS+1)-UXO(IS)))
1*(CRAY(II+1, KK)-CRAY(II, KK))
  EZ(I, 1, K)=EZ2+((Z(K)-UZ(KS-1))/(UZ(KS)-UZ(KS-1)))
1*(EZ1-EZ2)
```

290 CONTINUE

C

```
DO 300 I=1, LONE
DO 300 K=1, LTWO
  CRAY(I, K)=ARRAY(LL)
  LL=LL+1
```

300 CONTINUE

C

```
DO 310 I=1,28
  II=INT((I-2)/EXPFAC)+1
DO 310 K=1,28
  KK=INT((K+1)/EXPFAC)+1
  IS=II+INEAR-1
  KS=KK+KNEAR-1
  EZ1=CRAY(II, KK+1)+((XO(I)-UXO(IS))/(UXO(IS+1)-UXO(IS)))
1*(CRAY(II+1, KK+1)-CRAY(II, KK+1))
  EZ2=CRAY(II, KK)+((XO(I)-UXO(IS))/(UXO(IS+1)-UXO(IS)))
1*(CRAY(II+1, KK)-CRAY(II, KK))
  EZ(I, 29, K)=EZ2+((Z(K)-UZ(KS-1))/(UZ(KS)-UZ(KS-1)))
1*(EZ1-EZ2)
```

310 CONTINUE

C

RETURN
END

3.5.10 Subroutine DATASAV

Subroutine DATASAV is called by the main program DRIVER for each loop ($M=1,2$) selected (by MM setting) as frequently as output values of the induced surface charges ($\hat{n} \cdot \vec{E}$) and currents ($\hat{n} \times \vec{H}$) at the selected test points are required (determined by IP). A flow chart for the $M=1$ loop is given to Figure 15.

On the first call in a particular loop to Subroutine DATASAV parameters that describe the test points are set. These parameters are defined in Table 6. For this and all subsequent calls within the loop, DATASAV performs the operations described in the following paragraphs for each of the three field components from which the charge and the two current components are calculated. These operations allow one to define an arbitrary test point location and orientation where current and charge responses can be found from interpolated and extrapolated field responses at the appropriate surrounding I,J,K locations of the calculated fields.

Table 6. Variables Input in Subroutine DATASAV

Variable NAME	Description
NPTS	Number of test points
LOC(NPT)	Test point number for the M^{th} test point
NPLANE(NPT)	Scattering Plane indicates the orientation of the plane on which the M^{th} test point is located 1 - Normal is in the +x direction 2 - Normal is in the -x direction 3 - Normal is in the +y direction 4 - Normal is in the -y direction 5 - Normal is in the +z direction 6 - Normal is in the -z direction
XOBS(NPT)	x coordinate of the M^{th} test point
YOBS(NPT)	y coordinate of the M^{th} test point
ZOBS(NPT)	z coordinate of the M^{th} test point
THETA(NPT)	Measurement orientation in degrees from one of the principle axes of the problem space

ORIGINAL PAGE IS
OF POOR QUALITY

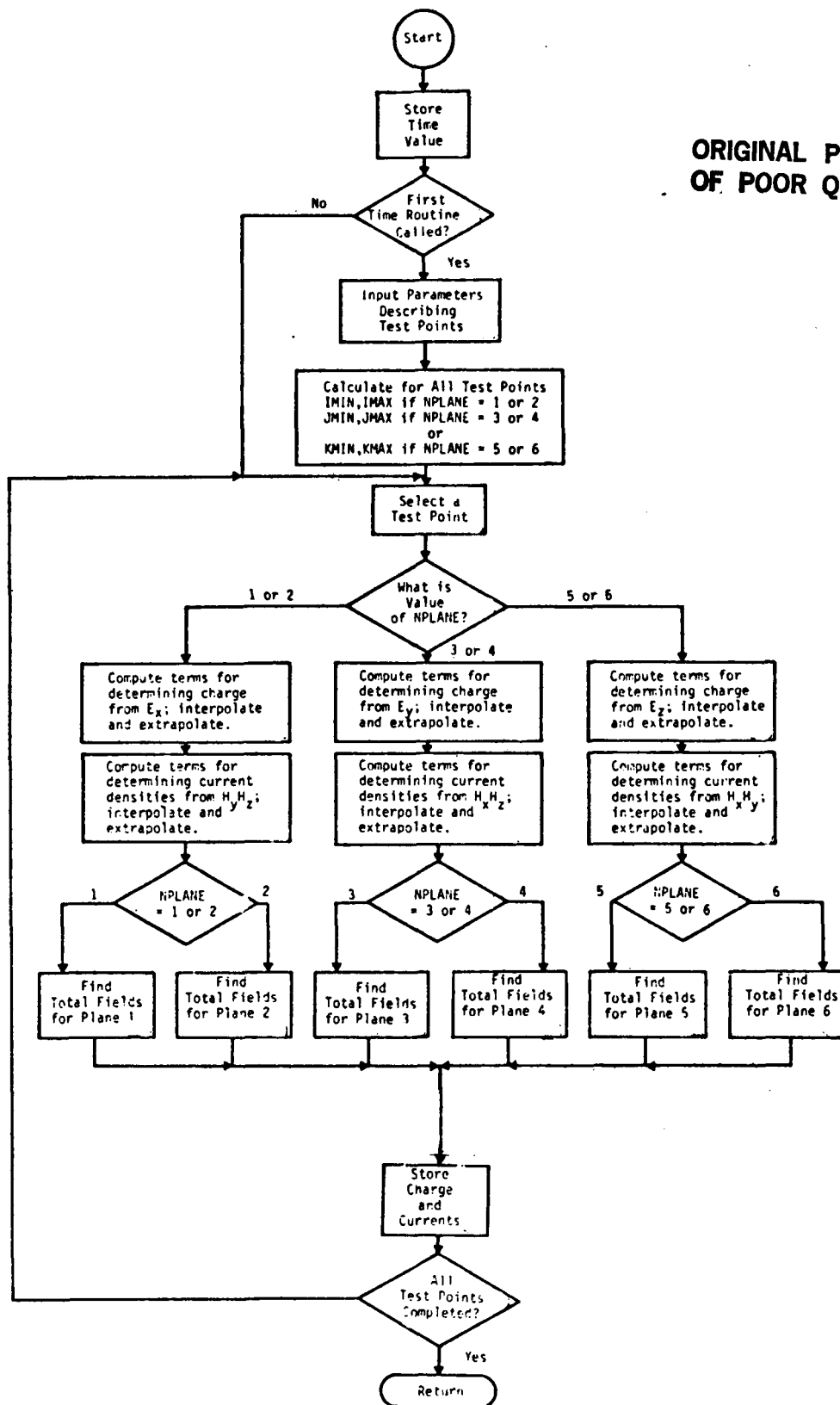


Figure 14. Flow Chart for Subroutine DATASAV

Using the test point location and the scattering plane orientation, eight field component locations are selected. These field component locations are distributed about the test point as illustrated below in Figure 16.

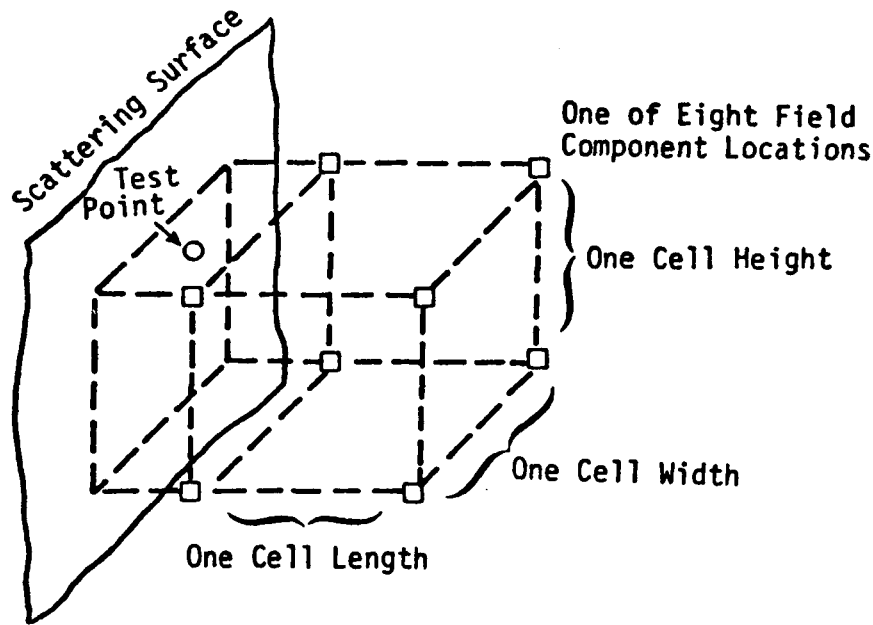


Figure 15. Distribution of Field Component Locations About Test Point

The selection process compared all the field component locations against the test point location in one of the three dimensions using a criterion of the form

if $XOBS(NPT) \leq X(I+1)$, then $I_{max} = (I+1)$ and $I_{min} = (I)$

(here the x coordinate has been used for illustrative purposes only), and in the remaining dimensions using a criterion of the form

if $YOBS(NPT) \leq Y(J+1)$, the $J_{min} = (J)$ and $J_{max} = (J+1)$.

By letting the coordinates by $X(I)$ or $X_O(I)$, $Y(J)$ or $Y_O(J)$, and $Z(K)$ or $Z_O(K)$ in the above criteria, depending on the field component selected, the test point is surrounded by the eight field components.

The eight field component values associated with the eight field component locations are given by:

$$\begin{aligned}
&\xi(I_{\min}, J_{\min}, K_{\min}) \\
&\xi(I_{\min}, J_{\min}, K_{\max}) \\
&\xi(I_{\min}, J_{\max}, K_{\min}) \\
&\xi(I_{\min}, J_{\max}, K_{\max}) \\
&\xi(I_{\max}, J_{\min}, K_{\min}) \\
&\xi(I_{\max}, J_{\min}, K_{\max}) \\
&\xi(I_{\max}, J_{\max}, K_{\min}) \\
&\xi(I_{\max}, J_{\max}, K_{\max})
\end{aligned}$$

where ξ is any one of the six field components. From these field components the value of $\xi(\text{XOBS}(\text{NPT}), \text{YOBS}(\text{NPT}), \text{ZOBS}(\text{NPT}))$ can be found by a combination of interpolation and extrapolation. The exact order in which the components are combined depends on the test point location and NPLANE. An example using the field component arrangement shown in Figure 17 is given to illustrate the process.

First an interpolation is performed in the z direction to yield

$$\begin{aligned}
\xi_1 &= \xi(I_{\min}, J_{\min}, K_{\min}) + (\xi(I_{\min}, J_{\min}, K_{\max}) \\
&\quad - \xi(I_{\min}, J_{\min}, K_{\min})) \cdot \frac{\Delta z}{\Delta Z} \\
\xi_2 &= \xi(I_{\max}, J_{\min}, K_{\min}) + (\xi(I_{\max}, J_{\min}, K_{\max}) \\
&\quad - \xi(I_{\max}, J_{\min}, K_{\min})) \cdot \frac{\Delta z}{\Delta Z} \\
\xi_3 &= \xi(I_{\min}, J_{\max}, K_{\min}) + (\xi(I_{\min}, J_{\max}, K_{\max}) \\
&\quad - \xi(I_{\min}, J_{\max}, K_{\min})) \cdot \frac{\Delta z}{\Delta Z} \\
\xi_4 &= \xi(I_{\max}, J_{\max}, K_{\min}) + (\xi(I_{\max}, J_{\max}, K_{\max}) \\
&\quad - \xi(I_{\max}, J_{\max}, K_{\min})) \cdot \frac{\Delta z}{\Delta Z}
\end{aligned}$$

These interpolated values are distributed about the test point as shown in Figure 18. They are then interpolated in the x direction to yield

$$\xi_{12} = \xi_1 + (\xi_2 - \xi_1) \frac{\Delta x}{\Delta X}$$

$$\xi_{34} = \xi_3 + (\xi_4 - \xi_3) \frac{\Delta x}{\Delta X}$$

Finally, the values ξ_{12} and ξ_{34} , as shown in Figure 19, are extrapolated to the test point location to give $\xi(\text{XOBS}(\text{NPT}), \text{YOBS}(\text{NPT}), \text{ZOBS}(\text{NPT}))$, using the expression

$$\xi(\text{XOBS}(\text{NPT}), \text{YOBS}(\text{NPT}), \text{ZOBS}(\text{NPT})) = \xi_{12} - (\xi_{34} - \xi_{12}) \frac{\Delta y}{\Delta Y}.$$

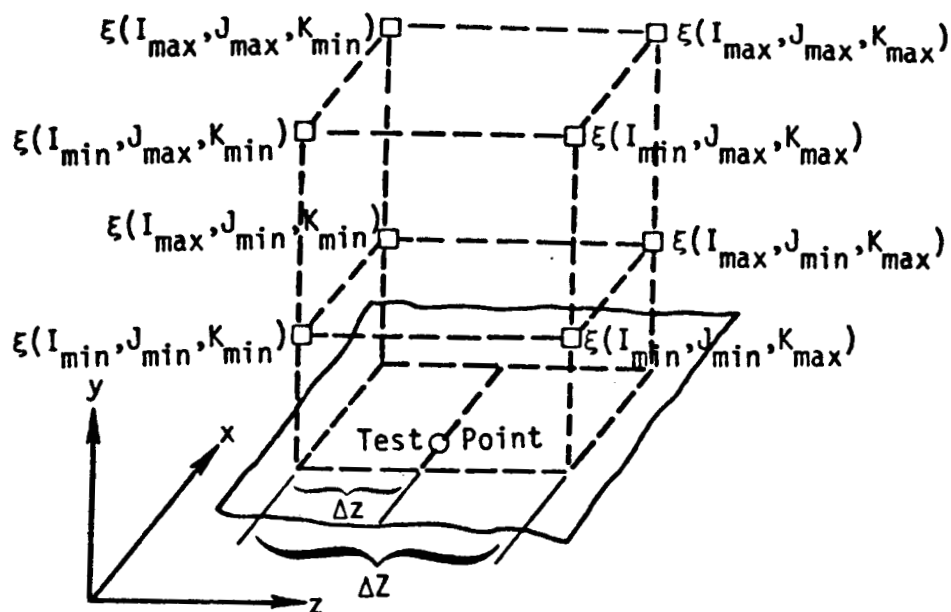


Figure 16. Example Field Component Arrangement about a Test Point

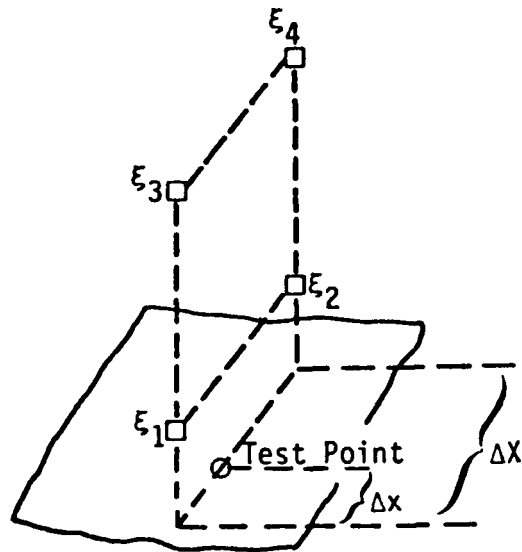


Figure 17. Interpolated Field Values about a Test Point

To each value of the scattered field component at the test location, $\xi(\text{XOBS}(\text{NPT}), \text{YOBS}(\text{NPT}), \text{ZOBS}(\text{NPT}))$, there is added the corresponding incident field component at the test point location. This is accomplished by adding the appropriate function value of EINCXP , EINCYP , EINCZP , HINCXP or HINCZP . The resulting total field components correspond to the induced surface charges and currents.

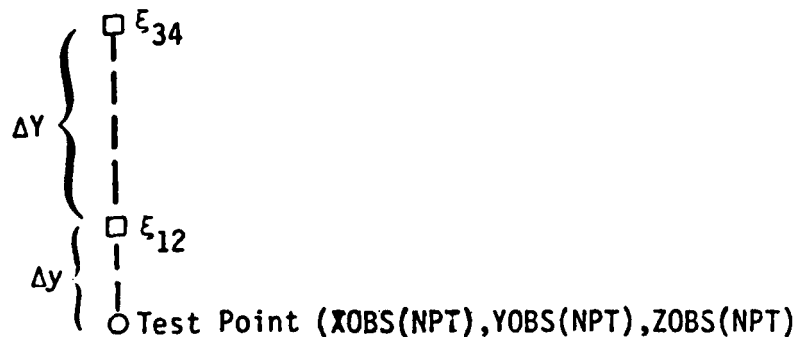


Figure 18. Interpolated Field Values Used to Extrapolate to Test Point Location

Two current components are calculated on any surface of the scatterer. These components can be combined to yield other orthogonal current component pairs corresponding to those measuring in an experiment where the current sensors are not aligned in the direction of the x, y or z axis. This situation is commonly the case for measurements made on aircraft wings where the current sensors are aligned with the axis of the wing, which generally is canted with respect to one of the principal axes of the problem space.

The angle THETA(NPT) is the sensor rotation at test point with respect to one of the principal axes. It is shown graphically in Figure 20. Simply stated, θ is always defined as falling in the first quadrant of the plane in which the scattering surface falls. As a result of this definition, the two rotated, orthogonal, current components, called H1 and H2, are always given by:

$$H1 = A \cos \theta + B \sin \theta$$

$$H2 = -A \sin \theta + B \cos \theta$$

where

$$A = H_z, B = H_y \text{ for NPLANE} = 1 \text{ or } 2$$

$$A = H_z, B = H_x \text{ for NPLANE} = 3 \text{ or } 4$$

$$A = H_x, B = H_y \text{ for NPLANE} = 5 \text{ or } 6$$

Those variables used locally in Subroutine DATASAV are defined in Table 7.

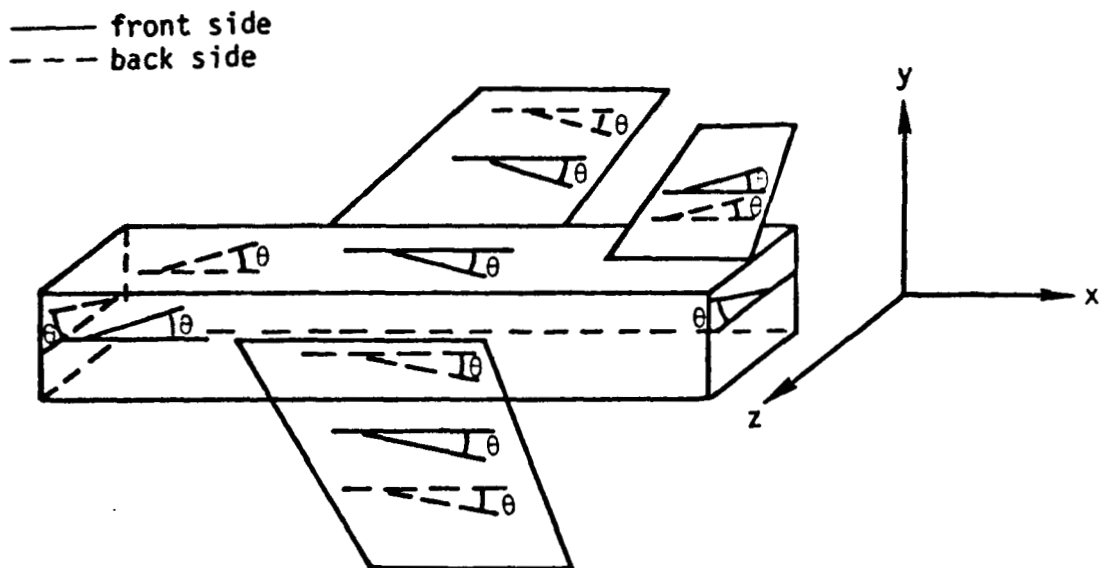


Figure 19. Graphical Description of THETA
(Angle defining sensor rotation).

Table 7. Local Variables - Subroutine DATASAV

Variable Name	Description
A	A principle H field component
B	A principle H field component
COSAN	Cosine of THETA
EX1,EX2 EX3,EX4 EX12,EX34 }	Interpolated E_x field components
EY1,EY2 EY3,EY4 EY12,EY34 }	Interpolated E_y field components
EZ1,EZ2 EZ3,EZ4 EZ12,EZ34 }	Interpolated E_z field components
HX1,HX2 HX3,HX4 HX12,HX34 }	Interpolated H_x field components
HY1,HY2, HY3,HY4 HY12,HY34 }	Interpolated H_y field components
HZ1,HZ2 HZ3,HZ4 HZ12,HZ34 }	Interpolated H_z field components
L	Counter on the number of time steps output
LOC	Test point number (array dimensioned on the number of test points)
NPLANE	Scattering Plane (array dimensioned on the number of test points)
NPTS	Number of test points
IM,IX JM,JX KM,KX }	Values of $I_{min}, I_{max}, J_{min}, J_{max}, K_{min}, K_{max}$
IMIN,IMAX JMIN,JMAX KMIN,KMAX }	Same as above (arrays dimensioned on the number of test points)
SINAN	Sine of THETA
TERM1,TERM2 TERM3,TERM4 TERM5,TERM6 TERM7,TERM8 TERM9 }	Interpolation and extrapolation terms

Table 7. (continued)

Variable Name	Description
THETA	Angle (degrees) of current sensor rotation (array dimensional on the number of test points)
XBS } YBS } ZBS }	Coordinates of test point location
XOBS } YOBS } ZOBS }	Same as above (arrays dimensioned on the number of test points)

SUBROUTINE DATASAV

C

```
COMMON/EFIELD / EX(29,29,29), EY(29,29,29), EZ(29,29,29)
COMMON/HFIELD / HX(29,29,29), HY(29,29,29), HZ(29,29,29)
COMMON/EXTRAS/NX,NY,NZ,NX1,NY1,NZ1,N,M,MQ,DT,XMU,EPSO,EPS,NPTLIM,
1 NN,NPTS,LMAX,SIGMA,C,T,PI,EXPFAC,IP,TX,TY,TZ,AMP,ALPHA,BETA,IDLS
COMMON/GRID/X(28),Y(28),Z(28),X0(29),Y0(29),Z0(29),
1 DX(29),DY(29),DZ(29),DX0(28),DY0(28),DZ0(28),
2 DXI(29),DYI(29),DZI(29),DXOI(28),DYOI(28),DZOI(28)
COMMON/OUT/ESTORE(250,24),HSTOR1(2500,24),HSTOR2(250,24),
1 TSTORE(250)
COMMON/CUMMUL/CUMM(24),LOC(24)
```

C

```
DIMENSION XOBS(30),YOBS(30),ZOBS(30)
DIMENSION SDX0(28),SDY0(28),SDZ0(28)
DIMENSION IMIN(30),IMAX(30),JMIN(30),JMAX(30),KMIN(30),KMAX(30)
DIMENSION THETA(30),NPLANE(30)
DIMENSION SX0(28),SY0(28),SZ0(28),SX(28),SY(28),SZ(28)
DIMENSION SEX(28,28,28),SEY(28,28,28),SEZ(28,28,28)
```

C

```
PIF=3.1415926536/180.0
L=N/IP
TSTORE(L)=T
```

C

```
IF(L.NE.1) GO TO 1120
IF(M.NE.1) GO TO 200
LOC(1)=1
LOC(2)=2
LOC(3)=3
LOC(4)=4
LOC(5)=5
LOC(6)=6
NPLANE(1)=3
NPLANE(2)=4
NPLANE(3)=1
NPLANE(4)=1
NPLANE(5)=1
NPLANE(6)=1
XOBS(1)=-1.8
YOBS(1)=0.0
ZOBS(1)=0.0
XOBS(2)=-4.8
YOBS(2)=-1.92
ZOBS(2)=0.0
XOBS(3)=-1.95
YOBS(3)=-1.68
ZOBS(3)=0.42
XOBS(4)=-1.95
YOBS(4)=-1.62
ZOBS(4)=0.48
XOBS(5)=-1.95
YOBS(5)=-1.68
ZOBS(5)=0.54
XOBS(6)=-1.95
YOBS(6)=-1.74
```

ZOBS(6)=0.48
THETA(1)=0.0
THETA(2)=0.0
THETA(3)=90.0
THETA(4)=0.0
THETA(5)=90.0
THETA(6)=0.0
NPTS=6
GO TO 499

C

200 CONTINUE
IF(M.NE.2)GO TO 300
LOC(1)=7
LOC(2)=8
LOC(3)=9
LOC(4)=10
LOC(5)=11
LOC(6)=12
NPLANE(1)=3
NPLANE(2)=4
NPLANE(3)=1
NPLANE(4)=1
NPLANE(5)=1
NPLANE(6)=1
XOBS(1)=-1.8
YOBS(1)=0.0
ZOBS(1)=0.0
XOBS(2)=-4.8
YOBS(2)=-1.92
ZOBS(2)=0.0
XOBS(3)=-1.95
YOBS(3)=-1.68
ZOBS(3)=0.42
XOBS(4)=-1.95
YOBS(4)=-1.62
ZOBS(4)=0.48
XOBS(5)=-1.95
YOBS(5)=-1.68
ZOBS(5)=0.54
XOBS(6)=-1.95
YOBS(6)=-1.74
ZOBS(6)=0.48
THETA(1)=0.0
THETA(2)=0.0
THETA(3)=90.0
THETA(4)=0.0
THETA(5)=90.0
THETA(6)=0.0
NPTS=6
GO TO 499

C

300 CONTINUE

C

C

DETERMINE CELLS FOR EXTRAPOLATION

C


```

DO 1110 NPT=1,NPTS
C
  IF(NPLANE(NPT).GE.3) GO TO 1025
  DO 1000 I=1,NX1
  IF(XOBS(NPT).LT.X(I+1)) GO TO 1020
1000 CONTINUE
1020 CONTINUE
  IF(NPLANE(NPT).EQ.1) IMIN(NPT) = I+1
  IF(NPLANE(NPT).EQ.2) IMIN(NPT) = I-1
  IMAX(NPT) = IMIN(NPT) + 1
  GO TO 1110
1025 CONTINUE
  IF(NPLANE(NPT).GT.4) GO TO 1070
  DO 1040 J=1,NY1
  IF(YOBS(NPT).LT.Y(J+1)) GO TO 1060
1040 CONTINUE
1060 CONTINUE
  IF(NPLANE(NPT).EQ.3) JMIN(NPT) = J+1
  IF(NPLANE(NPT).EQ.4) JMIN(NPT) = J-1
  JMAX(NPT) = JMIN(NPT) + 1
  GO TO 1110
1070 CONTINUE
  DO 1080 K=1,NZ1
  IF(ZOBS(NPT).LT.Z(K+1)) GO TO 1100
1080 CONTINUE
1100 IF(NPLANE(NPT).EQ.5) KMIN(NPT) = K+1
  IF(NPLANE(NPT).EQ.6) KMIN(NPT) = K-1
  KMAX(NPT) = KMIN(NPT) + 1
1110 CONTINUE
  PRINT 407,(XOBS(NPT),NPT=1,NPTS)
  PRINT 408,(YOBS(NPT),NPT=1,NPTS)
  PRINT 409,(ZOBS(NPT),NPT=1,NPTS)
C
C
C   NPLANE = 1 IS THE YZ PLANE, NORMAL AFT (TOWARD INCREASING I)
C
C   NPLANE = 2 IS THE YZ PLANE, NORMAL FORWARD(TOWARD DECREASING I)
C
C   NPLANE = 3 IS THE XZ PLANE, NORMAL UP (TOWARD INCREASING J)
C
C   NPLANE = 4 IS THE XZ PLANE, NORMAL DOWN(TOWARD DECREASING J)
C
C   NPLANE = 5 IS THE XY PLANE, NORMAL RIGHT (TOWARD INCREASING K)
C
C   NPLANE = 6 IS THE XY PLANE, NORMAL NEGATIVE (TOWARD DECREASING K)
C
1120 CONTINUE
  DO 1400 NPT=1,NPTS
  IM = IMIN(NPT)
  IX = IMAX(NPT)
  JM = JMIN(NPT)
  JX = JMAX(NPT)
  KM = KMIN(NPT)
  KX = KMAX(NPT)
  XBS = XOBS(NPT)

```

```

YBS = YOBS(NPT)
ZBS = ZOBS(NPT)
ANGLE = THETA(NPT)*PIF
C
C
IF(NPLANE(NPT).GT.4) GO TO 1270
IF(NPLANE(NPT).GT.2) GO TO 1155
C
C   NPLANE = 1 OR 2 -- YZ PLANES
C
C   COMPUTE TERMS FOR CHARGE
C
C       DETERMINE JM,JX,KM,KX  FOR EX
C
DO 1125 J=1,NY1
IF(YOBS(NPT).LT.YO(J+1)) GO TO 1128
1125 CONTINUE
1128 JM = J
JX = J+1
DO 1130 K=1,NZ1
IF(ZOBS(NPT).LT.ZO(K+1)) GO TO 1131
1130 CONTINUE
1131 KM = K
KX = K+1
IF(L.EQ.100) PRINT 410,IM,IX,JM,JX,KM,KX
TERM1 = (ZBS - ZO(KM))/DZO(KM)
C   TERM1 IS SCALE FACTOR FOR EX(I,J,KX)
TERM2 = (YBS - YO(JM))/DYO(JM)
C   TERM2 IS SCALE FACTOR FOR EX(I,JX,K)
TERM3 = (XBS - XO(IM))/DXO(IM)
TERM4 = 0.5*(DXO(IX)+DXO(IM))
TERM5 = 0.5*(DYO(JX)+DYO(JM))
TERM6 = 0.5*(DZO(KX)+DZO(KM))
C
EX1=EX(IM,JM,KM)+(EX(IM,JM,KX)-EX(IM,JM,KM))*TERM1
EX2=EX(IM,JX,KM)+(EX(IM,JX,KX)-EX(IM,JX,KM))*TERM1
EX3=EX(IX,JM,KM)+(EX(IX,JM,KX)-EX(IX,JM,KM))*TERM1
EX4=EX(IX,JX,KM)+(EX(IX,JX,KX)-EX(IX,JX,KM))*TERM1
EX12 = EX1 + (EX2-EX1)*TERM2
EX34 = EX3 + (EX4-EX3)*TERM2
C
C   DEBUG PRINT
C
IF(L.NE.100) GO TO 1134
PRINT 514,(((EX(I,J,K),I=IM,IX),J=JM,JX),K=KM,KX)
PRINT 515, EX1,EX2,EX3,EX4,EX12,EX34
1134 CONTINUE
C
C   COMPUTE TERMS FOR CURRENT DENSITY
C
C       DETERMINE KM,KX FOR HY,  JM,JX SAME AS FOR EX
C
DO 1135 K=1,NZ1

```

```

      IF(ZOBS(NPT).LT.Z(K+1)) GO TO 1136
1135 CONTINUE
1136 KM = K
      KX = K+1
      IF(L.EQ.100) PRINT 410,IM,IX,JM,JX,KM,KX
      TERM1 = (ZBS - ZO(KM))/DZO(KM)
      TERM2 = (YBS - YO(JM))/DYO(JM)
C      TERM2 IS SCALE FACTOR FOR HY(I,JX,K)
      TERM3 = (XBS - XO(IM))/DXO(IM)
      TERM4 = 0.5*(DXO(IX)+DXO(IM))
      TERM5 = 0.5*(DYO(JX)+DYO(JM))
      TERM6 = 0.5*(DZO(KX)+DZO(KM))
      TERM7 = (ZBS-Z(KM))/TERM6
C      TERM7 IS SCALE FACTOR FOR HY(I,J,KX)
      HY1=HY(IM,JM,KM)+(HY(IM,JM,KX)-HY(IM,JM,KM))*TERM7
      HY2=HY(IM,JX,KM)+(HY(IM,JX,KX)-HY(IM,JX,KM))*TERM7
      HY3=HY(IX,JM,KM)+(HY(IX,JM,KX)-HY(IX,JM,KM))*TERM7
      HY4=HY(IX,JX,KM)+(HY(IX,JX,KX)-HY(IX,JX,KM))*TERM7
C
      HY12 = HY1 + (HY2-HY1)*TERM2
      HY34 = HY3 + (HY4-HY3)*TERM2
C
C      DEBUG PRINT
C
      IF(L.NE.100) GO TO 1137
      PRINT 511,(((HY(I,J,K),I=IM,IX),J=JM,JX),K=KM,KX)
      PRINT 512, HY1, HY2, HY3, HY4, HY12, HY34
1137 CONTINUE
C
C
C      DETERMINE JM,JX,KM,KX FOR HZ
C
      DO 1138 J=1,NY1
      IF(YOBS(NPT).LT.Y(J+1)) GO TO 1140
1138 CONTINUE
1140 JM = J
      JX = J+1
      DO 1144 K=1,NZ1
      IF(ZOBS(NPT).LT.ZO(K+1)) GO TO 1146
1144 CONTINUE
1146 KM = K
      KX = K+1
      IF(L.EQ.100) PRINT 410,IM,IX,JM,JX,KM,KX
      TERM1 = (ZBS - ZO(KM))/DZO(KM)
C      TERM1 IS SCALE FACTOR FOR HZ(I,J,KX)
      TERM2 = (YBS - YO(JM))/DYO(JM)
      TERM3 = (XBS - XO(IM))/DXO(IM)
      TERM4 = 0.5*(DXO(IX)+DXO(IM))
      TERM5 = 0.5*(DYO(JX)+DYO(JM))
      TERM6 = 0.5*(DZO(KX)+DZO(KM))
C
      TERM8 = (YBS-Y(JM))/TERM5
C      TERM8 IS SCALE FACTOR FOR HZ(I,JX,K)
      HZ1=HZ(IM,JM,KM)+(HZ(IM,JM,KX)-HZ(IM,JM,KM))*TERM1
      HZ2=HZ(IM,JX,KM)+(HZ(IM,JX,KX)-HZ(IM,JX,KM))*TERM1

```

```

      HZ3=HZ(IX,JM,KM)+(HZ(IX,JM,KX)-HZ(IX,JM,KM))*TERM1
      HZ4=HZ(IX,JX,KM)+(HZ(IX,JX,KX)-HZ(IX,JX,KM))*TERM1
      HZ12 = HZ1 + (HZ2-HZ1)*TERM8
      HZ34 = HZ3 + (HZ4-HZ3)*TERM8
C
C          DEBUG PRINT
C
      IF(L.NE.100) GO TO 1148
      PRINT 504,(((HZ(I,J,K),I=IM,IX),J=JM,JX),K=KM,KX)
      PRINT 505, HZ1,HZ2,HZ3,HZ4,HZ12,HZ34
1148 CONTINUE
C
      IF(NPLANE(NPT).EQ.2) GO TO 1150
C
      TERM9 = (X(IM)-XBS)/TERM4
C          TERM9 IS -SCALE FACTOR FOR (EX,HY,HZ)(IX,J,K)
C
      ESTORE(L,NPT)=EX12 - (EX34-EX12)*TERM9 + EINCXP(XBS,YBS,ZBS)
      A=HZ12-(HZ34-HZ12)*TERM9+HINCZP(XBS,YBS,ZBS)
      B=HY12-(HY34-HY12)*TERM9+HINCYP(XBS,YBS,ZBS)
      GO TO 1375
C
C
C
1150 CONTINUE
      TERM9 = (XBS-X(IX))/TERM4
C          TERM9 IS -SCALE FACTOR FOR (EX,HY,HZ)(IM,J,K)
C
      ESTORE(L,NPT)=EX34+(EX34-EX12)*TERM9+EINCXP(XBS,YBS,ZBS)
      A=HZ34+(HZ34-HZ12)*TERM9+HINCZP(XBS,YBS,ZBS)
      B=HY34+(HY34-HY12)*TERM9+HINCYP(XBS,YBS,ZBS)
      GO TO 1375
C
C
C          NPLANE = 3 OR 4 -- XZ PLANES
C
C          COMPUTE TERMS FOR CHARGE
C
C          DETERMINE IM,IX,KM,KX FOR EY
C
1155 CONTINUE
      DO 1160 I=1,NX1
      IF(XOBS(NPT).LT.XO(I+1)) GO TO 1165
1160 CONTINUE
1165 IM = I
      IX = I+1
      DO 1170 K=1,NZ1
      IF(ZOBS(NPT).LT.ZO(K+1)) GO TO 1175
1170 CONTINUE
1175 KM = K
      KX = K+1
      IF(L.EQ.100)PRINT 410,IM,IX,JM,JX,KM,KX

```

```

      TERM1 = (ZBS - ZO(KM))/DZO(KM)
C      TERM1 IS SCALE FACTOR FOR EY(I,J,KX)
      TERM2 = (YBS - YO(JM))/DYO(JM)
      TERM3 = (XBS - XO(IM))/DXO(IM)
C      TERM3 IS SCALE FACTOR FOR EY(IX,J,K)
      TERM4 = 0.5*(DXO(IX)+DXO(IM))
      TERM5 = 0.5*(DYO(JX)+DYO(JM))
      TERM6 = 0.5*(DZO(KX)+DZO(KM))
      EY1=EY(IM,JM,KM)+(EY(IM,JM,KX)-EY(IM,JM,KM))*TERM1
      EY2=EY(IX,JM,KM)+(EY(IX,JM,KX)-EY(IX,JM,KM))*TERM1
      EY3=EY(IM,JX,KM)+(EY(IM,JX,KX)-EY(IM,JX,KM))*TERM1
      EY4=EY(IX,JX,KM)+(EY(IX,JX,KX)-EY(IX,JX,KM))*TERM1
C
      EY12 = EY1 + (EY2-EY1)*TERM3
      EY34 = EY3 + (EY4-EY3)*TERM3
C
C      DEBUG PRINT
C
      IF(L.NE.100) GO TO 1202
      PRINT 500,(((EY(I,J,K),I=IM,IX),J=JM,JX),K=KM,KX)
      PRINT 501, EY1,EY2,EY3,EY4,EY12,EY34
1202 CONTINUE
C
C      COMPUTE TERMS FOR CURRENT DENSITY
C
C      DETERMINE KM,KX FOR HX IM,IX SAME AS FOR EY
C
      DO 1210 K=1,NZ1
      IF(ZOBS(NPT).LT.Z(K+1)) GO TO 1220
1210 CONTINUE
1220 KM = K
      KX = K+1
      IF(L.EQ.100) PRINT 410,IM,IX,JM,JX,KM,KX
      TERM1 = (ZBS - ZO(KM))/DZO(KM)
      TERM2 = (YBS - YO(JM))/DYO(JM)
      TERM3 = (XBS - XO(IM))/DXO(IM)
C      TERM3 IS SCALE FACTOR FOR HX(IX,J,K)
      TERM4 = 0.5*(DXO(IX)+DXO(IM))
      TERM5 = 0.5*(DYO(JX)+DYO(JM))
      TERM6 = 0.5*(DZO(KX)+DZO(KM))
      TERM7 = (ZBS-Z(KM))/TERM6
C      TERM7 IS SCALE FACTOR FOR HX(I,J,KX)
      HX1=HX(IM,JM,KM)+(HX(IM,JM,KX)-HX(IM,JM,KM))*TERM7
      HX2=HX(IX,JM,KM)+(HX(IX,JM,KX)-HX(IX,JM,KM))*TERM7
      HX3=HX(IM,JX,KM)+(HX(IM,JX,KX)-HX(IM,JX,KM))*TERM7
      HX4=HX(IX,JX,KM)+(HX(IX,JX,KX)-HX(IX,JX,KM))*TERM7
C
      HX12 = HX1 + (HX2-HX1)*TERM3
      HX34 = HX3 + (HX4-HX3)*TERM3
C
C      DEBUG PRINT
C
      IF(L.NE.100) GO TO 1222
      PRINT 502,(((HX(I,J,K),I=IM,IX),J=JM,JX),K=KM,KX)

```

```

      PRINT 503, HX1,HX2,HX3,HX4,HX12,HX34
1222 CONTINUE
C
C
C      DETERMINE IM,IX,KM,KX FOR HZ
C
      DO 1225 I=1,NX1
      IF(XOBS(NPT).LT.X(I+1)) GO TO 1230
1225 CONTINUE
1230 IM = I
      IX = I+1
      DO 1235 K=1,NZ1
      IF(ZOBS(NPT).LT.ZO(K+1)) GO TO 1240
1235 CONTINUE
1240 KM = K
      KX = K+1
      IF(L.EQ.100) PRINT 410,IM,IX,JM,JX,KM,KX
      TERM1 = (ZBS - ZO(KM))/DZO(KM)
C      TERM1 IS SCALE FACTOR FOR HZ(I,J,KX)
      TERM2 = (YBS - YO(JM))/DYO(JM)
      TERM3 = (XBS - XO(IM))/DXO(IM)
      TERM4 = 0.5*(DXO(IX)+DXO(IM))
      TERM5 = 0.5*(DYO(JX)+DYO(JM))
      TERM6 = 0.5*(DZO(KX)+DZO(KM))
      HZ1=HZ(IM,JM,KM)+(HZ(IM,JM,KX)-HZ(IM,JM,KM))*TERM1
      HZ2=HZ(IX,JM,KM)+(HZ(IX,JM,KX)-HZ(IX,JM,KM))*TERM1
      HZ3=HZ(IM,JX,KM)+(HZ(IM,JX,KX)-HZ(IM,JX,KM))*TERM1
      HZ4=HZ(IX,JX,KM)+(HZ(IX,JX,KX)-HZ(IX,JX,KM))*TERM1
C
      TERM8 = (XBS-X(IM))/TERM4
C      TERM8 IS SCALE FACTOR FOR HZ(IX,J,K)
      HZ12 = HZ1 + (HZ2-HZ1)*TERM8
      HZ34 = HZ3 + (HZ4-HZ3)*TERM8
C
C      DEBUG PRINT
C
      IF(L.NE.100) GO TO 1242
      PRINT 504,(((HZ(I,J,K),I=IM,IX),J=JM,JX),K=KM,KX)
      PRINT 505, HZ1,HZ2,HZ3,HZ4,HZ12,HZ34
1242 CONTINUE
C
      IF(NPLANE(NPT).EQ.4) GO TO 1250
C
      TERM9 = (Y(JM)-YBS)/TERM5
C      TERM9 IS -SCALE FACTOR FOR (EY,HZ,HX)(I,JX,K)
C
      PRINT 995
      995 FORMAT(*CHK1*)
C
C      PRINT 996
      996 FORMAT(*CHK2*)
C
      ESTORE(L,NPT)=EY12 - (EY34-EY12)*TERM9 + EINCYP(XBS,YBS,ZBS)
      PRINT 997

```

```

997 FORMAT(*CHK3*)
C
  A = HZ12 - (HZ34-HZ12)*TERM9 + HINCZP(XBS,YBS,ZBS)
  PRINT 998
998 FORMAT(*CHK4*)
  B = HX12 - (HX34-HX12)*TERM9 + HINCXP(XBS,YBS,ZBS)
  PRINT 999
999 FORMAT(*CHK5*)
C
  C      DEBUG PRINT
C
  IF(L.NE.100) GO TO 1375
  EINC = EINCYP(XBS,YBS,ZBS)
  H1INC = HINCXP(XBS,YBS,ZBS)
  H2INC = HINCZP(XBS,YBS,ZBS)
  PRINT 513, EINC,H1INC,H2INC
  PRINT 506,ESTORE(L,NPT),A,B
  GO TO 1375
C
C
1250 TERM9 = (YBS-Y(JX))/TERM5
C      TERM9 IS -SCALE FACTOR FOR (EY,HZ,HX)(I,JM,K)
C
C
  ESTORE(L,NPT)=EY34 + (EY34-EY12)*TERM9 + EINCYP(XBS,YBS,ZBS)
  A = HZ34 + (HZ34-HZ12)*TERM9 + HINCZP(XBS,YBS,ZBS)
  B = HX34 + (HX34-HX12)*TERM9 + HINCXP(XBS,YBS,ZBS)
C
C      DEBUG PRINT
C
  IF(L.NE.100) GO TO 1375
  EINC = EINCYP(XBS,YBS,ZBS)
  H1INC = HINCXP(XBS,YBS,ZBS)
  H2INC = HINCZP(XBS,YBS,ZBS)
  PRINT 513, EINC,H1INC,H2INC
  PRINT 506,ESTORE(L,NPT),A,B
  GO TO 1375
C
C
  NPLANE = 5 OR 6 -- XY PLANES
C
  COMPUTE TERMS FOR CHARGE
C
C
  DETERMINE IM,IX,JM,JX FOR EZ
C
1270 CONTINUE
  DO 1275 I=1,NX1
  IF(XOBS(NPT).LT.XO(I+1)) GO TO 1280
1275 CONTINUE
1280 IM = I
  IX = I+1
  DO 1285 J=1,NY1
  IF(YOBS(NPT).LT.YO(J+1)) GO TO 1290
1285 CONTINUE

```

```

1290 JM=J
      JX = J+1
            IF(L.EQ.100) PRINT 410,IM,IX,JM,JX,KM,KX
      TERM1 = (ZBS - Z0(KM))/DZ0(KM)
      TERM2 = (YBS - Y0(JM))/DY0(JM)
C      TERM2 IS SCALE FACTOR FOR EZ(I,JX,K)
      TERM3 = (XBS - X0(IM))/DX0(IM)
C      TERM3 IS SCALE FACTOR FOR EZ(IX,J,K)
      TERM4 = 0.5*(DX0(IX)+DX0(IM))
      TERM5 = 0.5*(DY0(JX)+DY0(JM))
      TERM6 = 0.5*(DZ0(KX)+DZ0(KM))
1300 EZ1=EZ(IM,JM,KM)+(EZ(IX,JM,KM)-EZ(IM,JM,KM))*TERM3
      EZ2=EZ(IM,JX,KM)+(EZ(IX,JX,KM)-EZ(IM,JX,KM))*TERM3
      EZ3=EZ(IM,JM,KX)+(EZ(IX,JM,KX)-EZ(IM,JM,KX))*TERM3
      EZ4=EZ(IM,JX,KX)+(EZ(IX,JX,KX)-EZ(IM,JX,KX))*TERM3
C
      EZ12 = EZ1 + (EZ2-EZ1)*TERM2
      EZ34 = EZ3 + (EZ4-EZ3)*TERM2
C
C      DEBUG PRINT
C
      IF(L.NE.100) GO TO 1302
      PRINT 509,(((EZ(I,J,K),I=IM,IX),J=JM,JX),K=KM,KX)
      PRINT 510, EZ1,EZ2,EZ3,EZ4,EZ12,EZ34
1302 CONTINUE
C
C      COMPUTE TERMS FOR CURRENT DENSITY
C
C
C      DETERMINE JM,JX FOR HX , IM,IX SAME AS FOR EZ
C
      DO 1310 J=1,NY1
      IF(YOBS(NPT).LT.Y(J+1)) GO TO 1315
1310 CONTINUE
1315 JM = J
      JX = J+1
            IF(L.EQ.100) PRINT 410,IM,IX,JM,JX,KM,KX
      TERM1 = (ZBS - Z0(KM))/DZ0(KM)
      TERM2 = (YBS - Y0(JM))/DY0(JM)
      TERM3 = (XBS - X0(IM))/DX0(IM)
C      TERM3 IS SCALE FACTOR FOR HX(IX,J,K)
      TERM4 = 0.5*(DX0(IX)+DX0(IM))
      TERM5 = 0.5*(DY0(JX)+DY0(JM))
      TERM6 = 0.5*(DZ0(KX)+DZ0(KM))
      HX1=HX(IM,JM,KM)+(HX(IX,JM,KM)-HX(IM,JM,KM))*TERM3
      HX2=HX(IM,JX,KM)+(HX(IX,JX,KM)-HX(IM,JX,KM))*TERM3
      HX3=HX(IM,JM,KX)+(HX(IX,JM,KX)-HX(IM,JM,KX))*TERM3
      HX4=HX(IM,JX,KX)+(HX(IX,JX,KX)-HX(IM,JX,KX))*TERM3
C
      TERM7 = (YBS-Y(JM))/TERM5
C      TERM7 IS SCALE FACTOR FOR HX(I,JX,K)
      HX12 = HX1 + (HX2-HX1)*TERM7
      HX34 = HX3 + (HX4-HX3)*TERM7
C
C      DEBUG PRINT
C

```



```

C
  IF(L.NE.100) GO TO 1318
  PRINT 502,(((HX(I,J,K),I=IM,IX),J=JM,JX),K=KM,KX)
  PRINT 503, HX1,HX2,HX3,HX4,HX12,HX34
1318 CONTINUE
C
C
C      DETERMINE IM,IX,JM,JX FOR HY
C
  DO 1320 I=1,NX1
  IF(XOBS(NPT).LT.X(I+1)) GO TO 1325
1320 CONTINUE
1325 IM= I
  IX = I+1
  DO 1330 J=1,NY1
  IF(YOBS(NPT).LT.YO(J+1)) GO TO 1335
1330 CONTINUE
1335 JM = J
  JX = J+1
  IF(L.EQ.100) PRINT 410,IM,IX,JM,JX,KM,KX
  TERM1 = (ZBS - ZO(KM))/DZO(KM)
  TERM2 = (YBS - YO(JM))/DYO(JM)
C      TERM2 IS SCALE FACTOR FOR HY(I,JX,K)
  TERM3 = (XBS - XO(IM))/DXO(IM)
  TERM4 = 0.5*(DXO(IX)+DXO(IM))
  TERM5 = 0.5*(DYO(JX)+DYO(JM))
  TERM6 = 0.5*(DZO(KX)+DZO(KM))
  TERM8 = (XBS-X(IM))/TERM4
C      TERM8 IS SCALE FACTOR FOR HY(IX,J,K)
  HY1=HY(IM,JM,KM)+(HY(IX,JM,KM)-HY(IM,JM,KM))*TERM8
  HY2=HY(IM,JX,KM)+(HY(IX,JX,KM)-HY(IM,JX,KM))*TERM8
  HY3=HY(IM,JM,KX)+(HY(IX,JM,KX)-HY(IM,JM,KX))*TERM8
  HY4=HY(IM,JX,KX)+(HY(IX,JX,KX)-HY(IM,JX,KX))*TERM8
C
  HY12 = HY1 + (HY2-HY1)*TERM2
  HY34 = HY3 + (HY4-HY3)*TERM2
C
C      DEBUG PRINT
C
  IF(L.NE.100) GO TO 1340
  PRINT 511,(((HY(I,J,K),I=IM,IX),J=JM,JX),K=KM,KX)
  PRINT 512, HY1,HY2,HY3,HY4,HY12,HY34
1340 CONTINUE
C
  IF(NPLANE(NPT).EQ.6) GO TO 1350
C
  TERM9 = (Z(KM)-ZBS)/TERM6
C      TERM9 IS -SCALE FACTOR FOR (EZ,HX,HY)(I,J,KX)
C
C
  ESTORE(L,NPT)=EZ12 - (EZ34-EZ12)*TERM9 + EINCZP(XBS,YBS,ZBS)
  A = HX12 - (HX34-HX12)*TERM9 + HINCXP(XBS,YBS,ZBS)
  B = HY12 - (HY34-HY12)*TERM9 + HINCYP(XBS,YBS,ZBS)
C
C      DEBUG PRINT

```

```

C      IF(L.NE.100) GO TO 1375
      EINC = EINCZP(XBS,YBS,ZBS)
      H1INC= HINCXP(XBS,YBS,ZBS)
      H2INC= HINCYP(XBS,YBS,ZBS)
      PRINT 513,EINC,H1INC,H2INC
      PRINT 506,ESTORE(L,NPT),A,B
      GO TO 1375

C
C
1350 TERM9 = (ZBS-Z(KX))/TERM6
C      TERM9 IS -SCALE FACTOR FOR (EZ,HX,HY)(I,J,KM)
C
C
      ESTORE(L,NPT)=EZ34 + (EZ34-EZ12)*TERM9 + EINCZP(XBS,YBS,ZBS)
      A = HX34 + (HX34-HX12)*TERM9 + HINCXP(XBS,YBS,ZBS)
      B = HY34 + (HY34-HY12)*TERM9 + HINCYP(XBS,YBS,ZBS)

C
C      DEBUG PRINT
C
      IF(L.NE.100) GO TO 1375
      EINC = EINCZP(XBS,YBS,ZBS)
      H1INC= HINCXP(XBS,YBS,ZBS)
      H2INC= HINCYP(XBS,YBS,ZBS)
      PRINT 513,EINC,H1INC,H2INC
      PRINT 506,ESTORE(L,NPT),A,B
      GO TO 1375

C
1375 CONTINUE
      SINAN = SIN(ANGLE)
      COSAN = COS(ANGLE)
      HSTOR1(L,NPT) = A*COSAN+B*SINAN
      HSTOR2(L,NPT) = B*COSAN-A*SINAN
      CUMM(NPT)=CUMM(NPT)+HSTOR1(L,NPT)

C
      IF(L.EQ.1) GO TO 1390
      IF(L.NE.100) GO TO 1400
1390 CONTINUE
      PRINT 507, HSTOR1(L,NPT),HSTOR2(L,NPT),ESTORE(L,NPT),ANGLE
      PRINT 508,SINAN,COSAN
1400 CONTINUE

C
C      FORMAT STATEMENTS
C
401 FORMAT(I5)
402 FORMAT(2I5,4F10.3)
407 FORMAT(7H1  X OBS,/,3(10(F8.2,4X),/))
408 FORMAT(7H0  Y OBS,/,3(10(F8.2,4X),/))
409 FORMAT(7H0  Z OBS,/,3(10(F8.2,4X),/))
410 FORMAT(10H0  IM  IX,8X,7HJM  JX,8X,7HKM  KX,/,3(2I5,5X))
500 FORMAT(6H0  EY ,8E12.4,/)
501 FORMAT(1H0,4X,3HEY1,9X,3HEY2,9X,3HEY3,9X,3HEY4,8X,4HEY12,8X,
1      4HEY34,/,6E12.4,/)
502 FORMAT(6H0  HX ,8E12.4,/)
503 FORMAT(1H0,4X,3HHX1,9X,3HHX2,9X,3HHX3,9X,3HHX4,8X,4HHX12,8X,

```

```

1      4HHX34,/,6E12.4,/)
504 FORMAT(6H0 HZ ,8E12.4,/)
505 FORMAT(1H0,4X,3HHZ1,9X,3HHZ2,9X,3HHZ3,9X,3HHZ4,8X,4HHZ12,8X,
1      4HHZ34,/,6E12.4,/)
506 FORMAT(9H0 E,H1,H2,5X,3E12.4)
507 FORMAT(16H0 H1,H2,E,ANGLE,,5X,4E12.4)
508 FORMAT(14H0 SIN(ANGLE) =,F12.3,5X,12HCOS(ANGLE) =,F12.3)
509 FORMAT(6H0 EZ ,8E12.4,/)
510 FORMAT(1H0,4X,3HEZ1,9X,3HEZ2,9X,3HEZ3,9X,3HEZ4,8X,4HEZ12,8X,
1      4HEZ34,/,6E12.4,/)
511 FORMAT(6H0 HY ,8E12.4,/)
512 FORMAT(1H0,4X,3HHY1,9X,3HHY2,9X,3HHY3,9X,3HHY4,8X,4HHY12,8X,
1      4HHY34,/,6E12.4,/)
513 FORMAT(1H0,8H EINCP =,E12.4,2X,8HH1INCP =,E12.4,2X,
1      8HH2INCP =,E12.4)
514 FORMAT(6H0 EX ,8E12.4,/)
515 FORMAT(1H0,4X,3HEY1,9X,3HEY2,9X,3HEY3,9X,3HEY4,8X,4HEY12,8X,
1      4HEY34,/,6E12.4,/)

```

C

```

RETURN
END

```

3.5.11 Subroutine PRINOUT

PRINOUT prints out the data saved in DATASAV. It also extends the data record for H-field loops that define wire currents. The data is summed over time, which averages out any resonances (these behave as $A_n e^{-\alpha_n t} \sin 1^t$ and when summed over several cycles yield approximately zero.) This leaves the sum of the decaying exponential response $A_o e^{-\alpha_o t} dt$, which is $S = \int_0^T A_o e^{-\alpha_o t} dt$. The decaying exponential response is the portion of the total response that tracks the driving function which decays as $e^{-\alpha_o t}$. Thus, α_o is known from the excitation source and A_o can be found from S, the numerically derived sum, and the expression $S = \int_0^T A_o e^{-\alpha_o t} dt$. With A_o and α_o known, the data record is then extended in time. This allows transforms to be made for data records as long as 50 μ sec or a lower frequency limit of 10 KHz in the spectrum.

```

SUBROUTINE PRINOUT
C
COMMON/EXTRAS/NX,NY,NZ,NX1,NY1,NZ1,N,M,MQ,DT,XMU,EPSO,EPS,NPTLIM,
1 NN,NPTS,LMAX,SIGMA,C,T,PI,EXPFAC,IP,TX,TY,TZ,AMP,ALPHA,BETA,IDLS
COMMON/GRID/X(28),Y(28),Z(28),XO(29),YO(29),ZO(29),
1 DX(29),DY(29),DZ(29),DXO(28),DYO(28),DZO(28),
2 DXI(29),DYI(29),DZI(29),DXOI(28),DYOI(28),DZOI(28)
COMMON/OUT/ESTORE(250,24),HSTOR1(2500,24),HSTOR2(250,24),
1 TSTORE(250)
COMMON/CUMMUL/CUMM(24),LOC(24)
COMMON/PERM/MM
DIMENSION CSTORE(2500,8)
REAL PP

C
C
C PRINT OUTPUT
C
DO 1500 NPT = 1,NPTS
C
L=N/IP
PP=IP
C
PRINT 24,LOC(NPT)
24 FORMAT(8H1 LOC = ,I3)
PRINT 27,L
27 FORMAT (80X/36X,I4,40X)
PRINT 23, (TSTORE(LL),LL=1,L,3)
23 FORMAT(1X,1P6E13.6,1X)
PRINT 27,L
PRINT 23, (ESTORE(LL,NPT),LL=1,L,3)
PRINT 27,L
PRINT 23, (HSTOR1(LL,NPT),LL=1,L,3)
PRINT 27,L
PRINT 23, (HSTOR2(LL,NPT),LL=1,L,3)
C
TT=T
C
DO 1501 LL=L,LMAX
TT=TT+DT*8.
HSTOR1(LL,NPT)=CUMM(NPT)*DT*8.*
1 (1.0/((1.0-EXP(-ALPHA*T))/ALPHA-(1.0-EXP(-BETA*T))/BETA))*
2 EXP(-ALPHA*TT)
1501 CONTINUE
C
1500 CONTINUE
C
IF(M.NE.1.OR.MM.NE.1) GO TO 1600
C
DO 1510 LL=1,LMAX
CSTORE(LL,1)=HSTOR1(LL,1)
CSTORE(LL,2)=HSTOR1(LL,2)
1510 CONTINUE
C
PRINT 24,LOC(1)
PRINT 23,(CSTORE(LL,1),LL=1,LMAX,3)

```

```

        PRINT 24,LOC(2)
        PRINT 23,(CSTORE(LL,2),LL=1,LMAX,3)
C
        GO TO 500
C
1600 CONTINUE
C
        IF(M.NE.2.OR.MM.NE.12) GO TO 1700
C
        DO 1610 LL=1,LMAX
        CSTORE(LL,1)=HSTOR1(LL,1)
        CSTORE(LL,2)=HSTOR1(LL,2)
1610 CONTINUE
C
        PRINT 24,LOC(1)
        PRINT 23,(CSTORE(LL,1),LL=1,LMAX,3)
        PRINT 24,LOC(2)
        PRINT 23,(CSTORE(LL,2),LL=1,LMAX,3)
C
        GO TO 500
C
1700 CONTINUE
C
        IF(M.NE.2) GO TO 1800
C
        DO 1710 LL=1,LMAX
        CSTORE(LL,1)=HSTOR1(LL,1)
        CSTORE(LL,2)=HSTOR1(LL,2)
1710 CONTINUE
C
        PRINT 24,LOC(1)
        PRINT 23,(CSTORE(LL,1),LL=1,LMAX,3)
        PRINT 24,LOC(2)
        PRINT 23,(CSTORE(LL,2),LL=1,LMAX,3)
C
        GO TO 500
C
1800 CONTINUE
C
500 CONTINUE
C
        RETURN
        END
/EOB

```

APPENDIX A

The governing equations for the lossy dielectric finite difference code are in the form

$$\epsilon \frac{\partial E_x^S}{\partial t} + \sigma E_x^S = \left(\frac{\partial H_z^S}{\partial y} - \frac{\partial H_y^S}{\partial z} \right) - \sigma E_x^i - (\epsilon - \epsilon_0) \frac{\partial E_x^i}{\partial t}$$

where the x component of the E-field is used, for example. Consider now a plate of fractional thickness T in the x direction, i.e. the plate is $T\Delta x$ thick, so that T is a fraction of the cell dimension in the desired direction. The effective difference equation for the cell in which the plate is present can be found from the average of the above difference equation for a region in which the plate is present ($\alpha=0$, $\epsilon=\epsilon_0$) and where there is only free space ($\sigma=0$, $\epsilon=\epsilon_0$). More precisely

$$\begin{aligned} (1-T) \left[\epsilon_0 \frac{\partial E_x^S}{\partial t} = \frac{\partial H_z^S}{\partial y} - \frac{\partial H_y^S}{\partial z} \right] + \\ T \left[\epsilon \frac{\partial E_x^S}{\partial t} + \sigma E_x^S = \frac{\partial H_z^S}{\partial y} - \frac{\partial H_y^S}{\partial z} - \sigma E_x^i - (\epsilon - \epsilon_0) \frac{\partial E_x^i}{\partial t} \right] = \\ [(1-T)\epsilon_0 + T\epsilon] \frac{\partial E_x^S}{\partial t} + \sigma T E_x^S = \left(\frac{\partial H_z^S}{\partial y} - \frac{\partial H_y^S}{\partial z} \right) - \sigma T E_x^i - (\epsilon - \epsilon_0) T \frac{\partial E_x^i}{\partial t} \end{aligned}$$

In explicit finite difference form this is expressed as

$$\begin{aligned} E_x^S(I, J, K)^{n+1/2} &= E_x^S(I, J, K)^{n-1/2} e^{-\sigma T \Delta t / \epsilon_{eff}} \\ &+ (1 - e^{-\sigma T \Delta t / \epsilon_{eff}})_X \\ &\left[-E_x^i(I, J, K)^n - \frac{\epsilon - \epsilon_0}{\sigma} \dot{E}_x^i(I, J, K)^n \right. \\ &+ \frac{H_z^S(I, J, K)^n - H_z^S(I, J-1, K)^n}{\sigma T (Y(J) - Y(J-1))} \\ &\left. - \frac{H_y^S(I, J, K)^n - H_y^S(I, J, K-1)^n}{\sigma T (Z(K) - Z(K-1))} \right] \end{aligned}$$

where $\epsilon_{eff} = [(1-T)\epsilon_0 + T\epsilon]$.

Similar expressions hold in the other directions for the thin plate formalism.

REFERENCES

1. Yee, K. S., "Numerical Solution of Initial Boundary Value Problems Involving Maxwell's Equations in Isotropic Media," IEEE Trans. Ant. & Prop., Vol. AP-14, pp. 302-307, May 1966.
2. Merewether, D. E., "Transient Currents Induced on a Metallic Body of Revolution by an Electromagnetic Pulse," IEEE Trans. EMC, Vol. EMC-13, pp. 41-44, May 1971.
3. Taylor, C. D.; Lam, D. H. and Shumpert, T. H., "Electromagnetic Pulse Scattering in Time-Varying Inhomogeneous Media," IEEE Trans. Ant. Prop., Vol. AP-17, No. 5, September 1969.
4. Lax, P. D., "Differential Equations, Difference Equations and Matrix Theory," Commun. Pure Appl. Math., Vol. 2, pp. 175-194, November 1958.
5. Holland, R., "THREDE: A Free-Field EMP Coupling and Scattering Code," Mission Research Corporation, AMRC-R-85, September 1976.
6. Kunz, K. S. and Lee, K. M., A Three-Dimensional Finite-Difference Solution of the External Response of an Aircraft to a Complex Transient EM Environment: Part 1 - The Method and Its Implementation; and Part 2 - Comparison of Predictions and Measurements," IEEE Transactions EMC, Vol. EMC-20, No. 2, May 1978.

1. Report No. NASA CR-166079		2. Government Accession No.		3. Recipient's Catalog No.	
4. Title and Subtitle Generalized Three-Dimensional Experimental Lightning Code (G3DXL) User's Manual				5. Report Date February 1986	
				6. Performing Organization Code	
7. Author(s) Karl S. Kunz				8. Performing Organization Report No. KAI-R-1	
9. Performing Organization Name and Address Kunz Associates Incorporated P.O. Box 14373 Albuquerque, NM 87191				10. Work Unit No.	
				11. Contract or Grant No. NAS1-16591	
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546				13. Type of Report and Period Covered Contractor Report 1981-1986	
				14. Sponsoring Agency Code 505-66-21-04	
15. Supplementary Notes Langley Technical Monitor: Felix L. Pitts					
16. Abstract A new electromagnetic coupling computer code is an integration of several finite-difference codes based on THREDE, an established electromagnetic computer modeling technique. For a given lightning model, the new integrated code will be capable of predicting, with better resolution than was hitherto available, EM fields caused by lightning at several locations outside or inside metal or composite aircraft. Increased resolution is obtained by making two computer runs: total (unexpanded) geometry and expanded geometry. The total response is found and then is expanded, allowing greater resolution in geometry and higher frequency response.					
17. Key Words (Suggested by Author(s)) Finite difference code, Lightning, LEMP, Computer modeling			18. Distribution Statement Subject category - 47		
19. Security Classif. (of this report) Unclassified		20. Security Classif. (of this page) Unclassified		21. No. of Pages 144	
				22. Price	